Cadence 设计手册

使用手册

本手册共分为三部分：第一部分分为四章，分别介绍 Cadence cdsSpice、virtuoso Editing、Diva 和 verilog。第二部分主要介绍 MEDICI。第三部分是附录部分，是对前两章的一个补充,并简要的介绍了寄生元件提取语句的语法。

第一章． Cadence cdsSPICE 的使用说明

Cadence cdsSPICE 也是众多使用 SPICE 内核的电路模拟软件之一。因此他在使用上会有部分同我们平时所用到的 PSPICE 相同。这里我将侧重讲一下它的一些特殊用法。

§1－1 进入 Cadence 软件包

一．在工作站上使用

在命令行中（提示符后，如：ZUEDA22>）键入以下命令

icfb&↙(回车键)，其中& 表示后台工作。Icfb 调出 Cadence 软件。

出现的主窗口如图 1-1-1 所示：

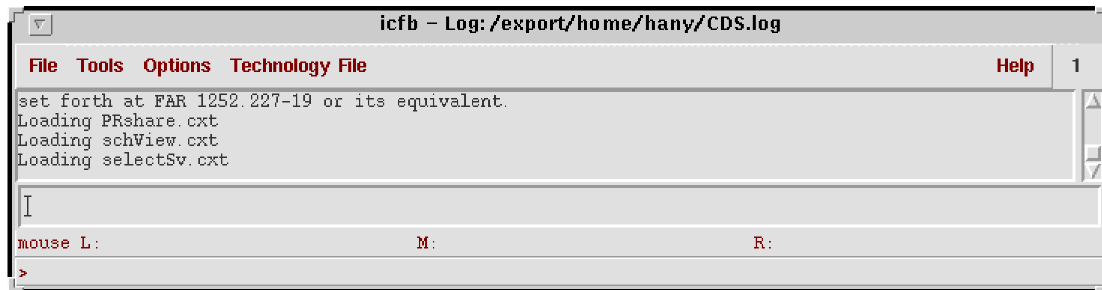


图 1-1-1Candence 主窗口

二．在 PC 机上使用

1）将 PC 机的颜色属性改为 256 色（这一步必须）；

2）打开 Exceed 软件，一般选用 xstart 软件，以下是使用步骤：

start method 选择 REXEC（TCP-IP），Programm 选择 Xwindow。Host 选择 10.13.71.32 或 10.13.71.33。host type 选择 sun。并点击后面的按钮，在弹出菜单中选择 command tool。确认选择完毕后，点击 run！

3）在提示符 ZDASIC22> 下键入:setenv DISPLAY 本机 ip:0.0(回车)

4）在命令行中（提示符后，如：ZUEDA22>）键入以下命令

icfb&↙(回车键)

即进入 cadence 中。出现的主窗口如图 1-1-1 所示。

以上是使用 xstart 登陆 cadance 的方法。在使用其他软件登陆 cadance 时，可能在登录前要修改文件.cshrc，方法如下：

在提示符下输入如下命令：vi .cshrc↙ （进入全屏幕编辑程序 vi）

将光标移至 setevn DISPLAY ZDASIC22:0.0 处，将“ZDASIC22”改为 PC 机的 IP，其它不变（重新回到服务器上运行时，还需按原样改回）。改完后存盘退出。

然后输入如下命令： source .cshrc↙ （重新载入该文件）

以下介绍一下全屏幕编辑程序 vi 的一些使用方法：

vi 使用了两种状态，一是**指令态**（Command Mode），另一是**插入态**（Insert Mode）。当 vi 处于指令态时，打入的内容会视作指令来解释；而当 vi 处于插入态时，就可以打入正文（text）文件；大多数 vi 指令是单字符的。由**插入态**改变为**指令态**，按〈Esc〉键；而由**命令态**转为**插入态**，则可以使用下面的插入令，直接打入，无需再按〈Return〉键。在 vi 的**指令态**下，用 h，j，k，l 键移动光标，具体如下：

h——光标左移一个字符；

j——光标向下一行；

k——光标向上一行；

l——光标右移一个字符；

以下是一些基本插入命令（须用到的）的用法：

i——在光标处插入正文；

x——删除光标处的字符；

:wq——存盘退出；

要记著一点，在插入态处，不能打入指令，必需先按〈Esc〉键，返回指令态。假若户不知身处何态，也可以按〈Esc〉键，不管处于何态，都会返回指令态其它的一些命令请读者自己参阅有关的书籍。

§1－2　建立可进行 SPICE 模拟的单元文件

主窗口分为信息窗口 CIW、命令行以及主菜单。信息窗口会给出一些系统信息（如出错信息，程序运行情况等)。在命令行中可以输入某些命令。如我们调用 Cadence 的命令 icfb 和一些其它命令，比较重要的有调出帮助文件的 openbook&等。

一．File 菜单

在 File 菜单下，主要的菜单项有 New、Open、Exit 等。在具体解释之前我们不妨先理顺一下以下几个关系。library(库)的地位相当于文件夹，它用来存放一整个设计的所有数据，像一些子单元（cell）以及子单元（cell）中的多种视图（view)。Cell（单元）可以是一个简单的单元，像一个与非门，也可以是比较复杂的单元（由 symbol 搭建而成)。View 则包含多种类型，常用的有 schamatic，symbol，layout，extracted，ivpcell 等等，他们各自代表什么意思以后将会一一提到。

New 菜单项的子菜单下有 Library、Cellview 两项。Library 项打开 New Library 窗口，Cellview 项打开 Create New File 窗口，如图 1-2-1 和 1-2-2 所示。

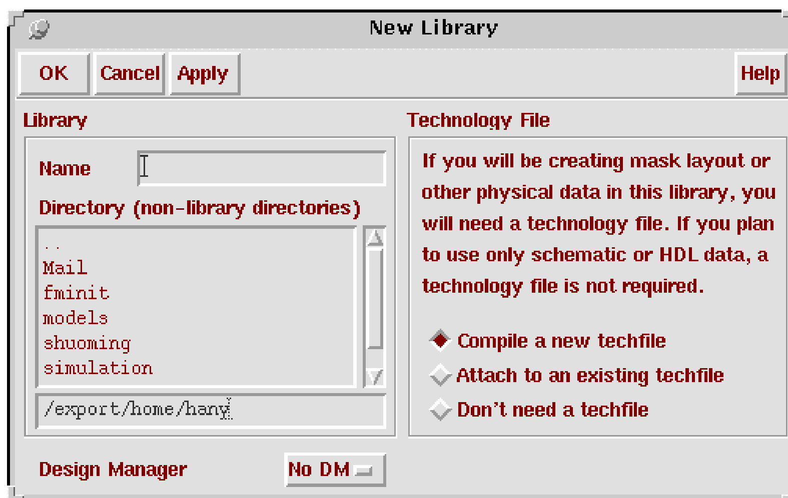


图 1-2-1　New Library 窗口

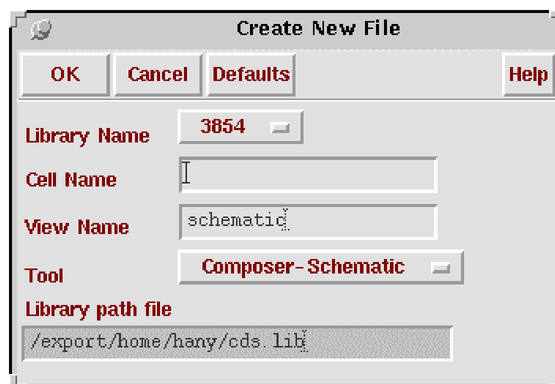


图 1-2-2 Create New File 窗口

1） 建立库(library):窗口分 Library 和 Technology File 两部分。Library 部分有 Name 和 Directory 两项，分别输入要建立的 Library 的名称和路径。如果只建立进行 SPICE 模拟的线路图，Technology 部分选择 Don't need a techfile 选项。如果在库中要创立掩模版或其它的物理数据（即要建立除了 schematic 外的一些 view），则须选择 Compile a new techfile(建立新的 techfile)或 Attach to an existing techfile(使用原有的 techfile)。

2） 建立单元文件(cell)：在 Library Name 中选择存放新文件的库，在 Cell Name 中输入名称，然后在 Tool 选项中选择 Composer-Schematic 工具(进行 SPICE 模拟)，在 View Name 中就会自动填上相应的 View Name——schematic。当然在 Tool 工具中还有很多别的工具，常用的象 Composer－symbol、virtuoso－layout 等，分别建立的是 symbol、layout 的视图（view）。在 Library path file 中，是系统自建的 library path file 文件的路径及名称(保存相关库的名称及路径)。

Open 菜单项打开相应的 Open File 窗口，如图 1-2-3 所示。

在 Library Name 中选择库名，在 Cell Names 中选择需要打开的单元名。Mode 项可以选择打开方式——可编辑状态或者只读状态。

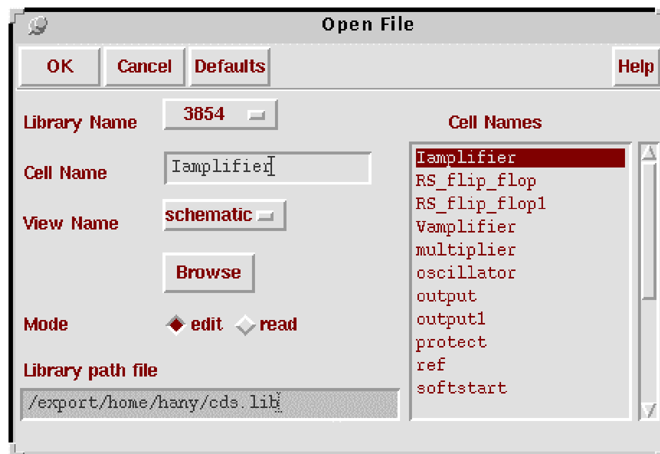


图 1-2-3 Open File 窗口

Exit 项退出 Cadence 软件包。

二．Tools 菜单

在 Tools 菜单下，主要的菜单项有 Library Manager、Library Path Editor 等。

Library Manager 项打开的是库管理器（Library Manager）窗口，如图 1-2-4 所示。

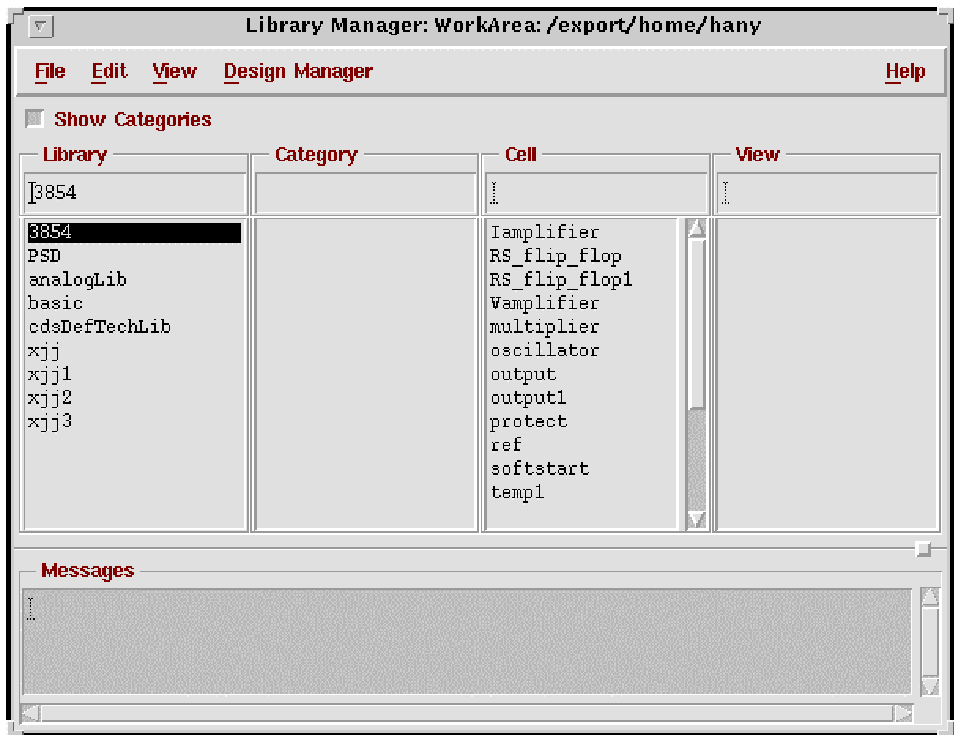


图 1-2-4 Library Manager 窗口

在窗口的各部分中，分别显示的是 Library、Category、Cell、View 相应的内容。双击需要打开的 view 名（或同时按住鼠标左右键从弹出菜单中选择 Open 项）即可以打开相应的文件。同样在 library manager 中也可以建立 library 和 cell。具体方法是点击 file，在下拉菜单中选择 library 或 cell 即可。

Library Path Editor 项打开的是 Library Path Editor 窗口，如图 1-2-5 所示。

从 File 菜单中选择 Add Library 项，填入相应的库名和路径名，即可包括入相应的库。

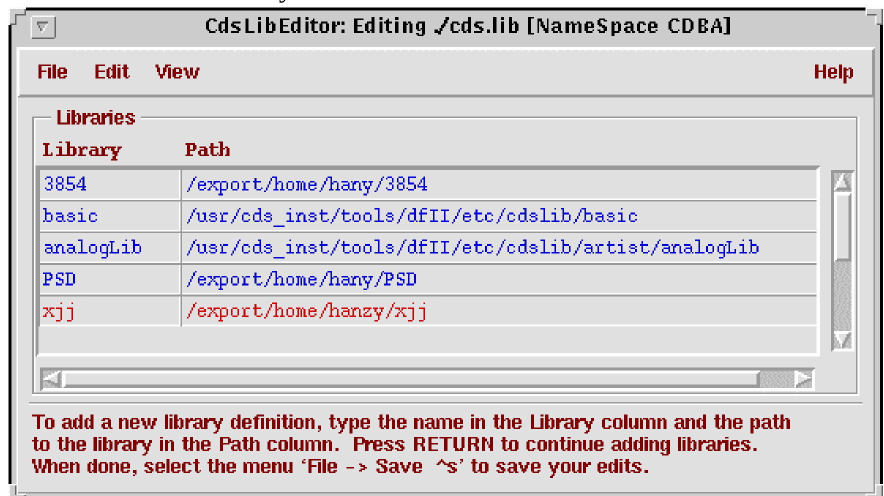


图 1-2-5 Library Path Editor 窗口

三．Technology File 菜单

这个菜单中的最后一项 Edit Layers 可以使用在版图编辑中，用来修改原始图层的一些属性。

§1—3 编辑可进行 SPICE 模拟的单元文件

选择主窗口的 File→Open→Open file，打开相应的文件，即进入了 Composer-Schematic Editing 窗口，如图 1-3-1 所示。窗口左边的按钮分别(从上到下)为 Check and Save（检查并存盘）、Save（存盘）、Zoom out by 2（放大两倍）、Zoom in by 2（缩小两倍）、Stretch（延伸）、Copy（拷贝）、Delete（删除）、Undo（取消）、Property（属性）、Component（加元件）、Wire(Narrow)（画细线）、Wire(Wide)（画粗线）、Pin（管脚）、Cmd options、Repeat（重复），这些分别可以在菜单中找到相应的菜单项。

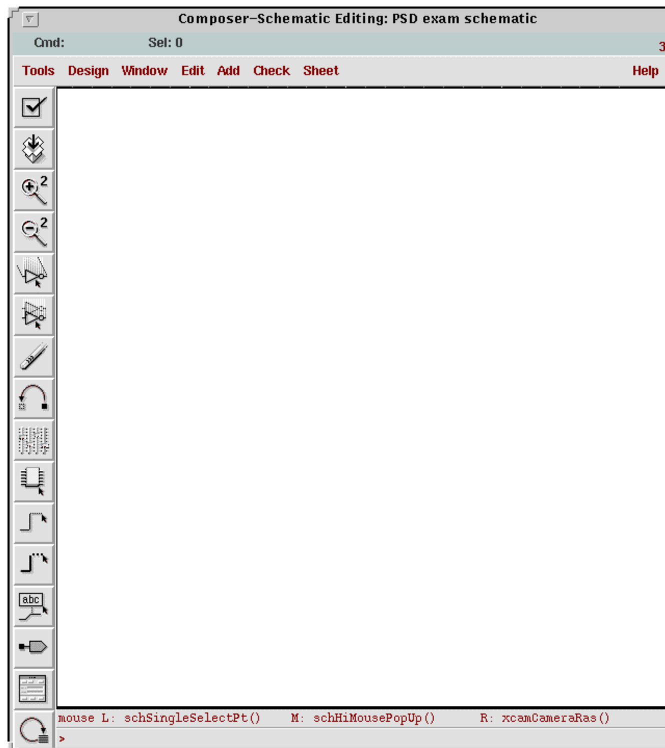


图 1-3-1 Composer-Schematic Editing 窗口

选择 Add/Component 菜单，打开相应添加元件的窗口，如图 1-3-2 所示。点击 Browse，会弹出 library manager 窗口，一些常用的元器件都在 Analoglib 库中。 View Name 一般选择 symbol，instance Names 不用自己填，系统会自己加上去 。添加完元件后需设定元件的模型名称（如果必须的话）以及一些参数的值，特别是 mos 管和三极管，一定要填 model name，

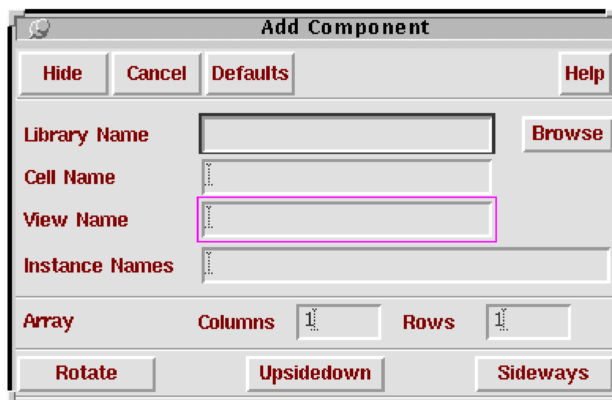


图 1-3-2 添加元件窗口

否则在模拟时会出错（我们一般使用华晶的元件 model）。填好后，就可以将元件添加到 Editing 的编辑窗口中去了。其它的一些连线、移动、删除、复制的操作和一般的 EDA 工具差不多，这儿就不一一再说了。还有一点要提到的是，对于交叉相连的两条线，系统会有警

告，可对连线稍作修改去除这个警告。

注：

以下是一些常用的快捷键：

i——添加元件，即打开添加元件的窗口；

[——缩小两倍；

]——扩大两倍；

w——连线（细线）；

f——全图显示；

p——查看元件属性。

从一种状态转为另一种状态，按 escape，或直接点击图标或使用快捷键。

为了使电路图更加明了，一般在电路的输入输出部分加上 pin 脚。这在后面的例子中将会提到。

§1—4　模拟的设置(重点)

Composer-schamatic 界面中的 Tools→Analog Artist 项可以打开 Analog Artist Simulation

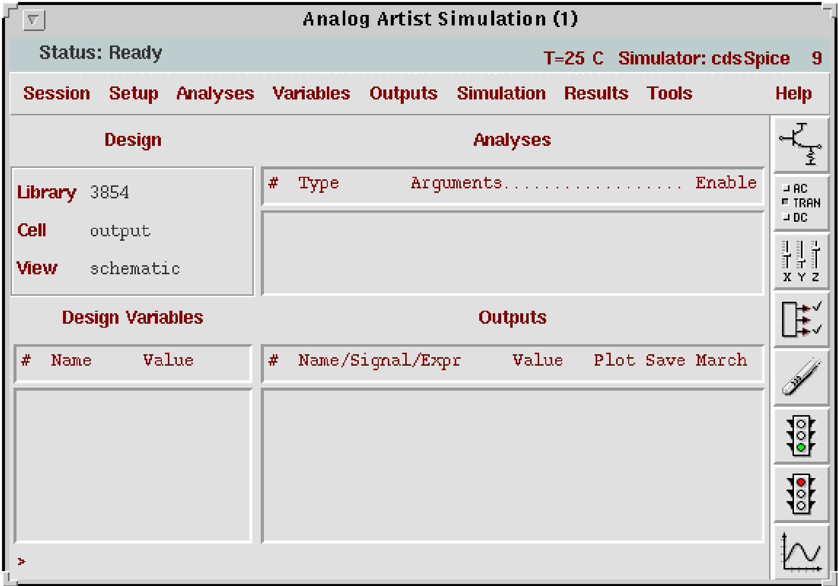


图 1-4-1 Analog Artist Simulation 窗口

窗口，如图 1-4-1 所示。这是模拟时用到的主要工具，接下去主要介绍一下有关的内容。

一．Session 菜单

包括 Schematic Window、Save State、Load State、Options、Reset、Quit 等菜单项。Schematic window 项回到电路图；Save State 项打开相应的窗口，保存当前所设定的模拟所用到的各种

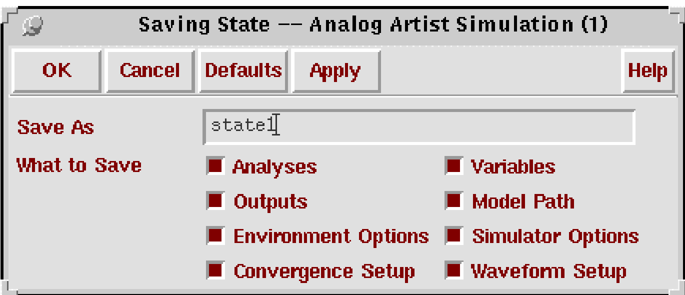


图 1-4-2　Save State 窗口

参数。如图 1-4-2 所示。窗口中的两项分别为状态名和选择需保存的内容。

Load State 打开相应的窗口，加载已经保存的状态。

Reset 重置 analog artist。相当于重新打开一个模拟窗口。

二．Setup 菜单

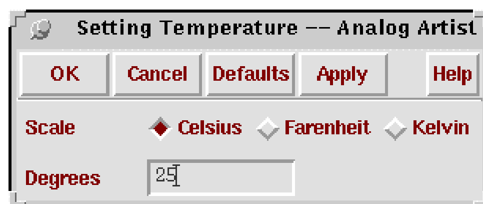
包括 Design、Simulator/directory/host、Temperature、Model Path 等菜单项：

Design 项选择所要模拟的线路图。

Simulator/directory/host 项选择模拟使用的模型，系统提供的选项有 cdsSpice、hspiceS、spectreS 等等。我们一般用到的是 cdsSpice 和 spectreS。其中采用 spectreS 进行的模拟更加精确。下面我们只以这两种工具为例说明。

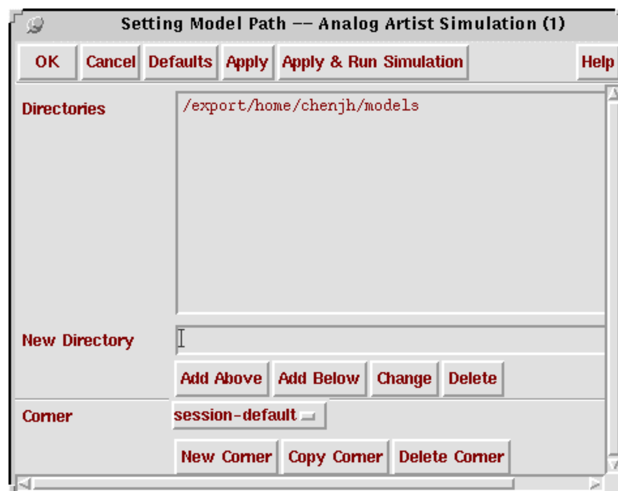
Temperature 打开如图 1-4-3 的窗口，可以设置模拟时的温度。

图 1-4-3　温度设置窗口



Model Path 打开如图 1-4-4 的窗口，设置元件模型的路径。系统会自动在所设定的路径下寻找器件 model name 对应的 model 模型。

图 1-4-4 模型路径设置窗口



三．Analyses 菜单

选择模拟类型。在 cdsSpice 下有 ac、dc、tran、noise 四个选项，分别对应的是交流分析、直流分析、瞬态分析和噪声分析。我们知道：交流分析是分析电流（电压）和频率之间的关系，因此在参数范围选择时是选择频率。直流分析是分析电流（电压）和电流（电压）间的关系。Tran 分析是分析参量值随时间变化的曲线。他们分别的窗口如下图所示。其设置很直观，这里就不在赘述。

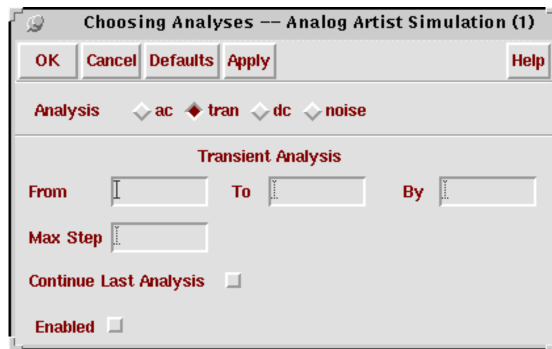


图 1-4-5 瞬态分析设置

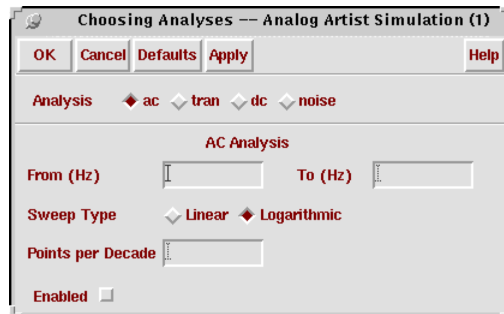


图 1-4-6 交流分析设置

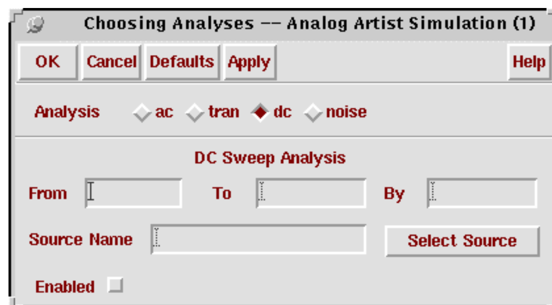
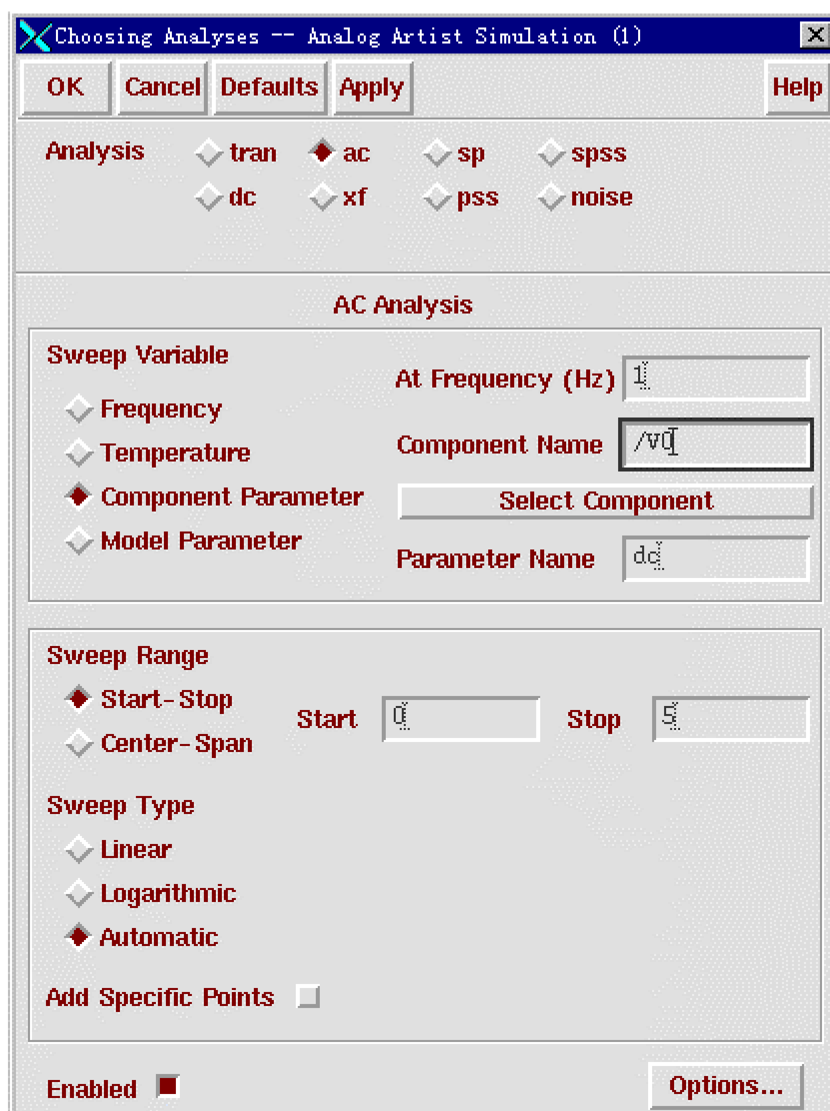
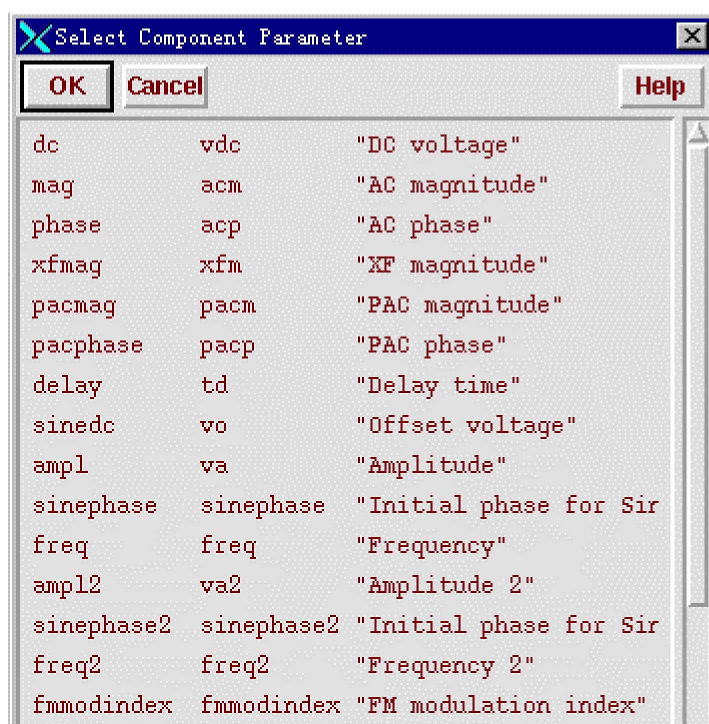


图 1-4-7 直流分析设置

而在 spectreS 中，可供选择的分析类型有很多，常用的还是 ac、dc、tran 和 noise，不过它们设置与 cdsSpice 不同。Tran 的设置只需填入模拟停止时间即可。 ac 和 dc 分析的设置则更具特点：spectreS 提供了**变量扫描功能**（和参量扫描有些类似），其中可供选择的变量（parameter）有 frequency（ac 分析）、temperature、component parameter 和 model parameter。以下一一说明：在 ac 分析扫描频率（常规分析）时，只需填入起始频率和终止频率即可。而在扫描其他参数时，必须将整个电路固定在一个工作频率（at frequency）上，然后进行其它选择。要进行 component parameter 扫描时，先点击 select component，然后在电路图上选择所需扫描的器件，这时会弹出一个列有可供扫描参量名称的菜单，在其上选择即可。进行 model parameter 扫描时只需填入 model name 和 parameter name 即可。当然，以上扫描都免不了要填写扫描范围，就不多说了。以下是一些图示：

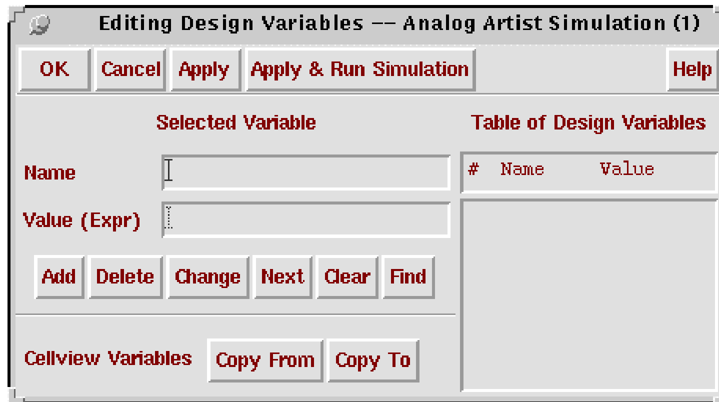

Choosing Analyses -- Analog Artist Simulation (1)
OK
Cancel
Defaults
Apply
Help
Analysis
tran
ac
sp
spss
dc
xf
pss
noise
AC Analysis
Sweep Variable
Frequency
Temperature
Component Parameter
Model Parameter
At Frequency (Hz)
1
Component Name
/V0
Select Component
Parameter Name
dc
Sweep Range
Start-Stop
Center-Span
Start
0
Stop
5
Sweep Type
Linear
Logarithmic
Automatic
Add Specific Points
Enabled
Options...


Select Component Parameter
OK
Cancel
Help
dc vdc "DC voltage"
mag acm "AC magnitude"
phase acp "AC phase"
xfmag xfm "XF magnitude"
pacmag pacm "PAC magnitude"
pacphase pacp "PAC phase"
delay td "Delay time"
sinedc vo "Offset voltage"
ampl va "Amplitude"
sinephase sinephase "Initial phase for Sir
freq freq "Frequency"
ampl2 va2 "Amplitude 2"
sinephase2 sinephase2 "Initial phase for Sir
freq2 freq2 "Frequency 2"
fmmodindex fmmodindex "FM modulation index"

四．**Variables** 菜单

包括 Edit 等子菜单项。Edit 项打开如图 1-4-5 的窗口。可以对变量进行添加、删除、查找、复制等操作。变量（variables）既可以是电路中元器件的某一个参量，也可以是一个表达式。变量将在参量扫描（parametric analysis）时用到，以下会提到。

图 1-4-5 变量编辑窗口



五．**其它有关**的菜单项

1）Tools/Parametric Analysis 子菜单可以打开如图 1-4-6 的窗口。它提供了一种很重要的分析方法——参量分析的方法，也即参量扫描。可以对温度，用户自定义的变量（variables）进行扫描，从而找出最合适的值。以下详细说明：

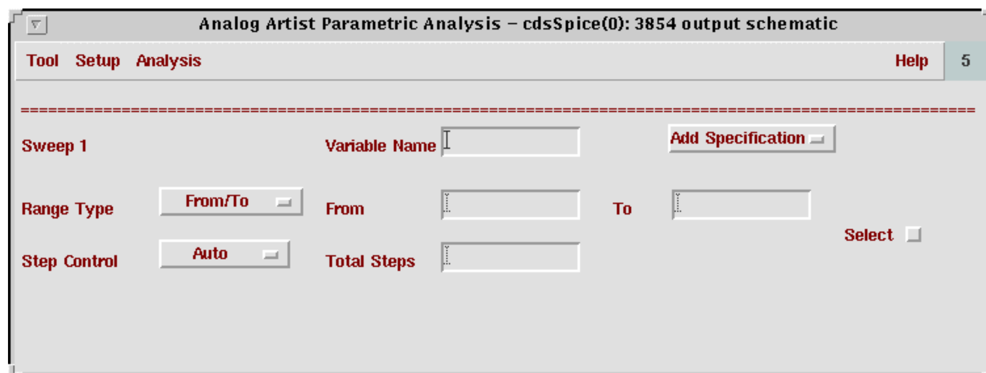


图 1-4-6 参量分析窗口

参量扫描

在模拟中，如果对某一元件的参数大小不确定，不知值取多大可以得到最优的结果时，可以将该参数设为变量，进行变量扫描，比较输出结果，从而确定参数的值。另外，对系统变量也可以进行扫描，如温度变量（temp）。

步骤：

a.在 Edit Variables 窗口中添加新的变量,如是对系统变量(如温度)扫描，就略去这一步；

b.在 Parametric Analysis 窗口(如图 1-4-5 所示)中，填入变量名称（温度变量是 temp），设定扫描范围以及步长等。也可以点击 setup，在 pick name for variables 的弹出菜单中选择所需扫描的参量（除系统参量外，菜单中所列举的都是 variables 中设置的变量）。其实这个工作和我们前面提到的 spectreS 中的变量扫描很象，不过它更加完备（因为可以对一个表达式进行扫描），所以读者应当将两种方法都掌握。

然后运行 Analysis 菜单下的 start 子菜单，开始模拟，模拟结果会在 Waveform 窗口中显示。

2） Outputs/To be plotted/selected on schematic 子菜单用来在电路原理图上选取要显示的波形（点击连线选取节点电压，点击元件端点选取节点电流），这个菜单比较常用。当然我们需要输出的有时不仅仅是电流、电压，还有一些更高级的。比如说：带宽、增益等需要计算的值，这时我们可以在 Outputs/setup 中设定其名称和表达式。在运行模拟之后，这些输出将会很直观的显示出来。举个例子：标识 3db 的点，我们用到的表达式如下：bandwidth（VF(“/Out)，3，“ low ”）。需要注意的是：表达式一般都是通过计算器（caculator）输入的。Cadance 自带的计算器功能强大，除了输入一些普通表达式以外，还自带有一些特殊表达式，如 bandwidth、average 等等。本文在最后会对计算器作介绍。

下面介绍一下 analog artist 窗口的情况，在 Analog Artist 窗口中靠右的一列按钮分别是：

Choose Design：选择模拟的电路；

Choose Analyses（选择模拟的类型）：瞬态模拟、直流模拟或交流模拟；

Edit Variables（变量编辑）：打开变量编辑窗口；

Setup Outputs：输出设置；

Delete：删除变量等；

Run Simulation：开始模拟；

Stop Simulation：停止模拟；

Plot Outputs：波形输出。

§ 1－5　模拟结果的显示以及处理

在模拟有了结果之后，如果设定的 output 有 plot 属性的话，系统会自动调出 waveform 窗口，并显示 outputs 的波形。如图 1-5-1 所示。

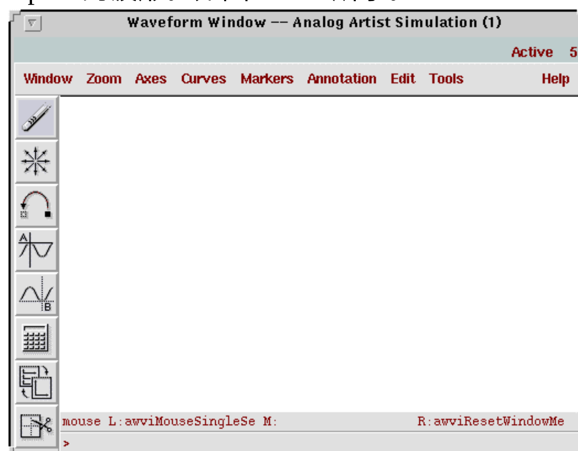


图 1-5-1　波形显示窗口

其左边的一列按钮分别为：

Delete（删除）：删除图中的某个波形；

Move（移动）：移动某个波形的位置，可以把几个波形叠加在一个坐标轴下；点击该按钮，然后点击需要移动的波形，再在目的地点击左键，即可完成移动操作；

Undo（取消）：取消前一次操作；

Crosshair MarkerA、 Crosshair MarkerB：十字标志 A 和 B；

Calculator（计算器）：计算器工具（可以对输出波形进行特定的处理 ）；

Switch Axis Mode（坐标轴模式切换）：同一坐标显示所有波形或分别在各自的坐标下显示；

Add Subwindow：添加子窗口。

§1－6　一个例子——D 触发器

1、 电路图的输入

这是一个带 R 清零端（低电平有效）的 D 触发器，由 20 个 MOS 管组成，其中 NMOS 管和 PMOS　管各为 10 个，组成四个传输门、两个反门和两个与非门。

具体的电路如图 1-6-1

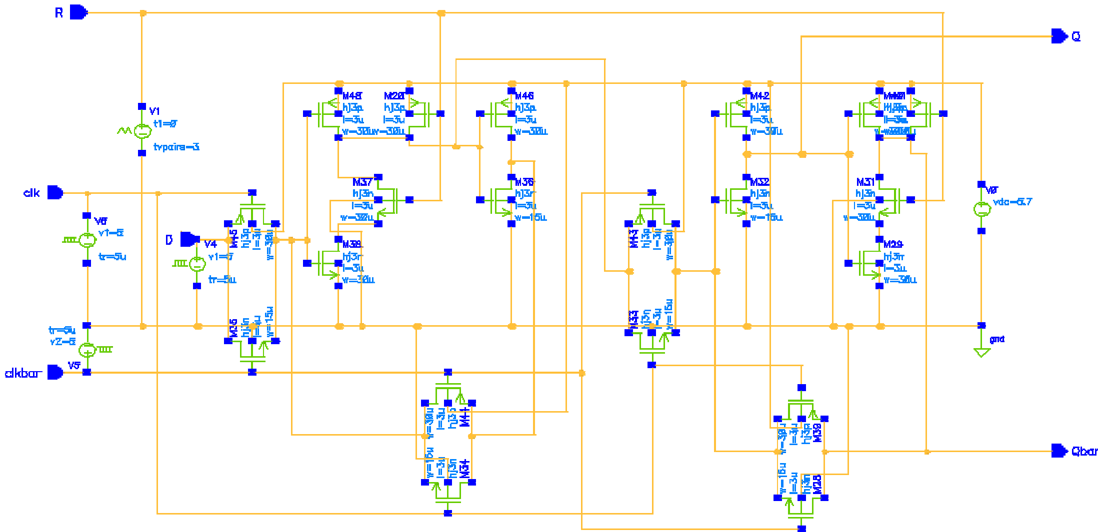


图 1-6-1　D 触发器电路图

D 触发器真值表

时钟（clk）	D	Q
0	X	Q
1	0	0
1	1	1

其中的一些参数设置如下：

传输门的 PMOS：W——30μ，L——3μ；　　model:hj3p(在 models 目录下)

NMOS：W——15μ，L——3μ；　　model:hj3n;

与非门的 PMOS：W——30μ，L——3μ；

NMOS：W——30μ，L——3μ；

非门的 PMOS：W——30μ，L——3μ；

NMOS：W——15μ，L——3μ；

电源直流电压:5.7V；

R 端的信号源(R):

cellname——vpwl;

Number of pairs of points——3（信号源波形上有三个转折点）；

Time 1——0s；

Voltage 1——0V;

Time 2——100μs；

Voltage 2——0V;

Time 3——105μs；

Voltage 3——5V;

Delay time——500ns;

时钟信号（clk）：

cellname——vpulse;
Voltage 1——0V;
Voltage 2——5V;
Delay time——5μs;
Rise time——5μs;
Pulse time——100μs;
Period time——200μs;

时钟信号的反（clkbar）：

cellname——vpulse;
Voltage 1——5V;
Voltage 2——0V;
Delay time——5μs;
Rise time——5μs;
Pulse time——100μs;
Period time——200μs;

D 端输入（D）：

cellname——vpulse;
Voltage 1——0V;
Voltage 2——5V;
Delay time——5μs;
Rise time——5μs;
Pulse time——100μs;
Period time——200μs;

瞬态分析设置如下：

From:0　　to:1ms　　by:1μs

得到的波形如图 1-6-2 所示：

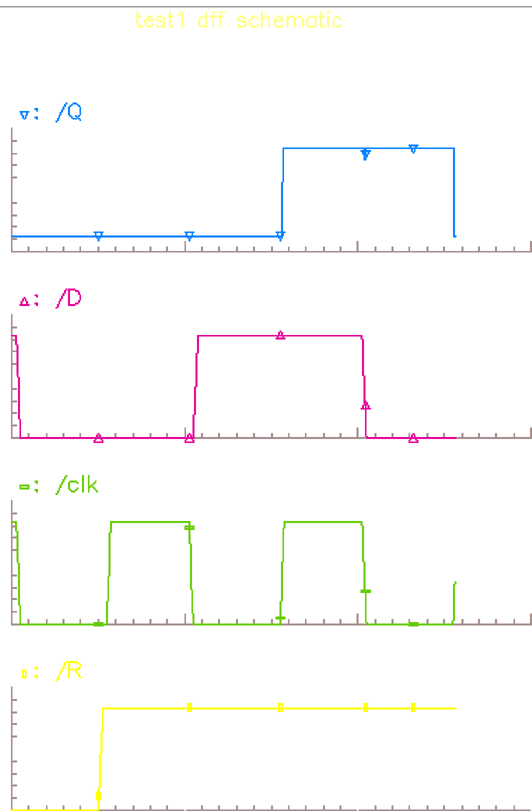


图 1-6-2　cdsSPICE 模拟结果 1

可以看到模拟的结果符合 D 触发器的逻辑。但是有一个问题出现了，注意到我们所设的时间是从 0→1ms，但是输出的模拟结果到 600μs 左右就截止了，这是和模拟的工具有关。

为了得到较好的模拟结果，可以换一种工具——spectreS 来完成模拟。

在 Analog Artist Simulation 窗口中选 Setup 下的 Simulator/directory/host 子菜单，出现如图 1-6-3 的设置窗口。在 Simulator 项中选择 spectreS 工具。然后在 Choosing Analyses 弹出的设置窗口中设定 stop time 为 1ms，模拟的结果如图 1-6-4 所示，将得到一个很好的结果。。

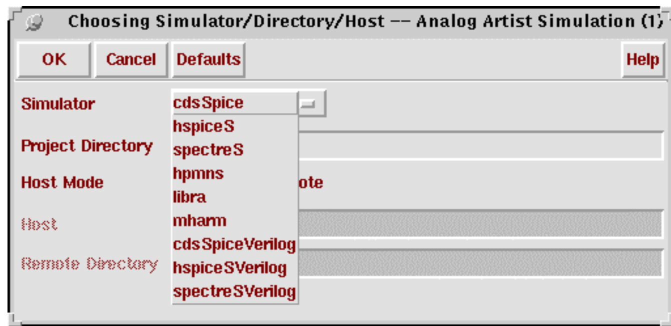


图 1-6-3 选择模拟工具窗口

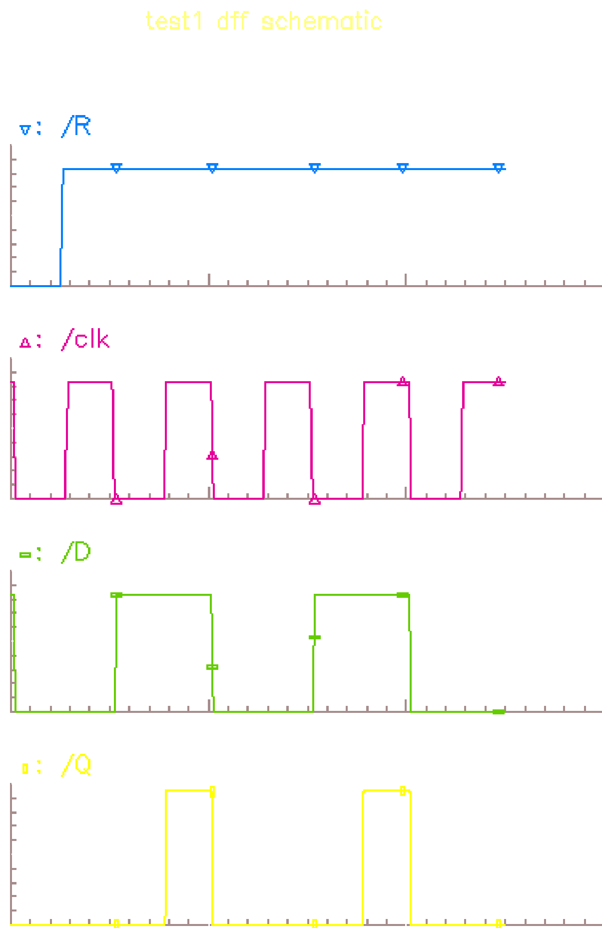


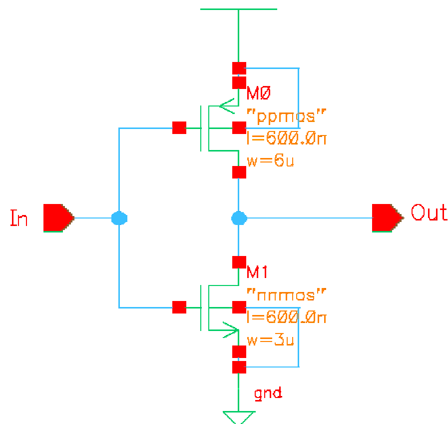
图 1-6-4　spedtreS 模拟结果

§ 1－7　分模块模拟(建立子模块)

在电路越来越复杂的情况下，如果再花时间去建立一个象 D 触发器这样复杂的 schamatic，明显会使工作更繁复。因此我们在建立了一个子电路后，可以将其看作一个整体，建立一个模块,即建立一个 symbol（view name），放在用户自己库里的作为一个器件(component)来用。

下面通过子模块非门的建立，来说明这一内容。

在 Library Manager 中分别建立非门 not(cell)的 schematic(view)和 symbol(view)，如图 1-7-1(a) 和 1-7-1(b)所示。两者的 PIN 的名称必须一致，这样才能建立起一一对应的关系。



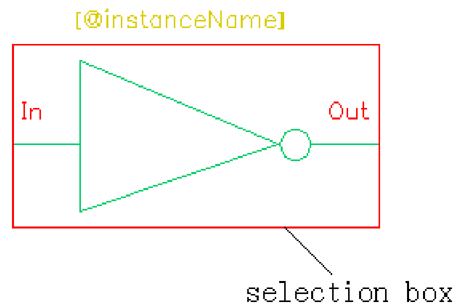


图 1-7-1(a)　　图 1-7-1(b)

建立 symbol(view)的步骤：

在 Library Manager 中新建 cell，在如图 1-2-2 的窗口的 Tool 项选择 Composer-symbol,即建立的是 symbol(view);

用子菜单 Add/Shape/Line 和 Add/Shape/Circle 的命令画出如右图的形状；

用子菜单 Add/label 的命令添加标签[@instanceName]；

用子菜单 Add/Selection Box 命令添加选择框。

另一种建立 symbol(view)的方法是：打开 not(cell)的 schematic(view)，用子菜单 Design/Create Cellview/From Cellview 命令。出现以下的窗口，如图 1-7-2

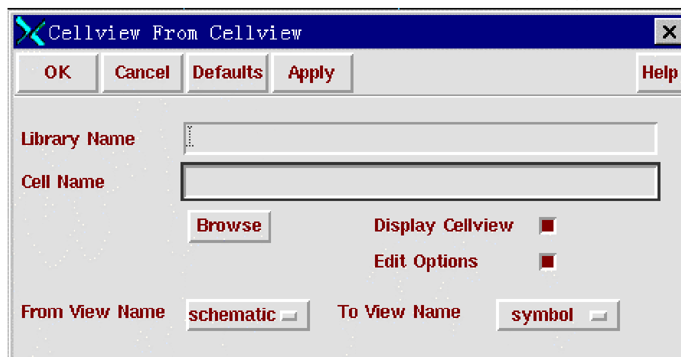
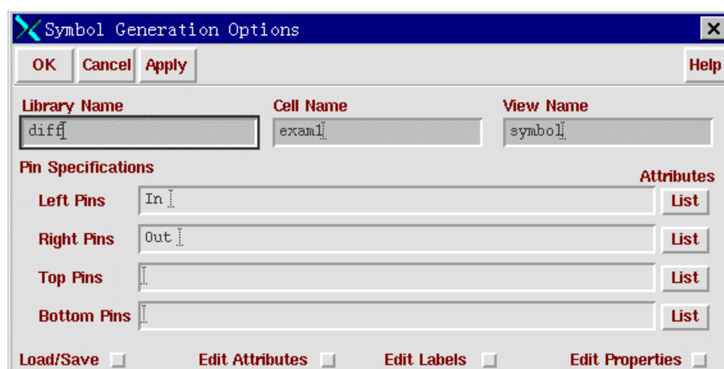


图 1-7-2　从一个 view 建立另一个 view

输入相应的名称后，单击 OK，就出现如图 1-7-3 的选项窗口。其建立的 symbol 如图 1-7-4 所示，如果不是建立有常用符号的子模块，如与门，非门等逻辑门，这种方法是较快



的。

图 1-7-3　建立 symbol 的选项窗口



图 1-7-4　第二种方法建立的 symbol 图形

这样就建立了一个最简单的子模块——非门。在模拟过程中，就可以通过添加元器件（component）来直接将非门加到电路中来，而不用具体画出其内部的结构，这实际上就是以一个简单的 symbol 来代替其内部的复杂结构。以此类推，可以将小模块一步步的拼凑成大的模块，直接用于模拟仿真。有一点要注意的是：对于有源器件（如非门）建立 symbol，必须在原始电路图上添加 analoglib 中的源和地，而且源的电压值也需要设定好，否则变为 symbol 搭成电路后会出错。当然用于模拟时设定的激励源是不用加在电路图中的

§1—8　其它的一些内容

计算器

计算器有两种格式，一种是代数格式，另一种 RPN（逆波兰）格式。有时需要对 Waveform 窗口中显示的波形进行处理，如改变坐标轴的单位（将电压单位改成分贝形式等），比较两个量的差值（显示两个电压的差）。所有的这些可以用 Calculator 工具来实现，如图 1-8-1 所示。

除了常规的计算以外，计算器还可以完成波形处理等工作。下面就简单地介绍一下常用的内容。

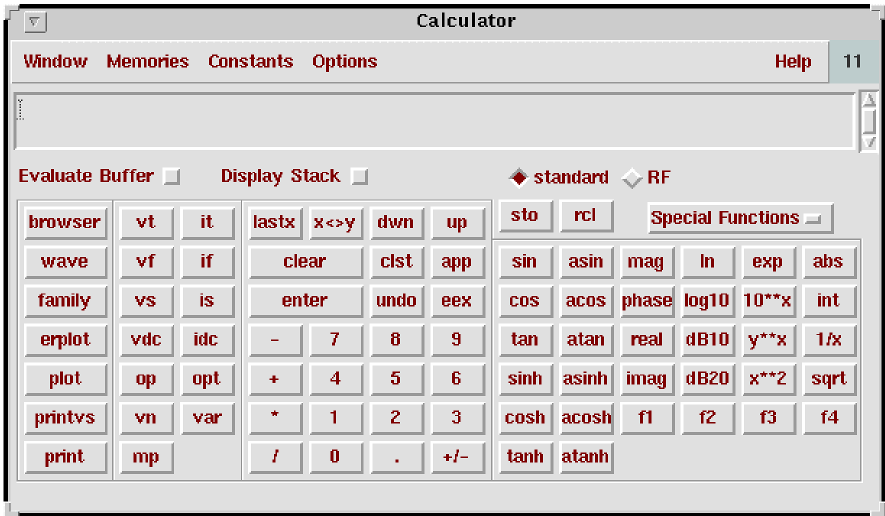


图 1-8-1 计算器工具

图 1-8-1 中显示的是逆波兰模式。菜单 Options/set Algebraic 或 set RPN 可以切换模式。Calculator 窗口中的按钮可以分为下面几个部分：

1. 功能键（选择、打印波形曲线，绘波形图）；
2. 常规计算器键盘；
3. 函数键。

下面分别介绍他们的功能。

一．功能键：

1. browser：打开结果浏览窗口（Result Browser）。它有如下作用：

。观察模拟波形和文本结果

。绘制波形

。将波形表达式直接拷入计算器窗口中

2.wave、family：从波形窗口（waveform Window）中选择所要处理的曲线波形。Wave 是选择单一的波形，family 是选择一组波形（如参数扫描得到的曲线簇）。

3.erplot、plot：在波形窗口（waveform Window）中绘制曲线波形。Erplot 是先擦除原先的波形，然后再绘出新的曲线波形；plot 是直接在原波形窗口中追加新的曲线波形。

4.printvs、print：打印曲线波形抑或是显示测量的数值。

5.电原理图表达式键：在电原理图中选择需要处理的数据（如电压、电流）具体如下表所示。

vt	瞬态电压	it	瞬态电流
vf	频率电压	if	频率电流
vs	源扫描电压	is	源扫描电流
vdc	直流电压	op	直流工作点
vn	噪声电压	opt	瞬态工作点
var	变量	mp	模型参数

二．常规计算器键盘：

这部分和常规计算器的键盘基本相同，除了少数几个键，如 undo 键。对于逆波兰模式，其输入形式需遵循逆波兰表达式的格式。先介绍几个键：lastx：上次 buffer（显示窗口）中的数值或变量、x<>y：buffer 中的值与 stack1（堆栈 1）的值互换、dwn：下压堆栈、up：堆栈弹出、clear：清除 buffer 中的值、clst：将 buffer 和 stack 中的所有值都清除。下面举个例子：输入（1＋x）/x。其输入步骤为：1，enter，clear，x，＋，lastx，/

三．函数键

1． 常规函数键：

如下表所示。

三角函数	Sin，cos，tan，sinh，cosh，tanh，asin，acos，atan，asinh，acosh，atanh	
其他常规函数	Mag	幅度
	phase	相位
	real	实部
	imag	虚部
	Ln、log10、dB10、dB20、exp、10**x、y**x、x**2、abs、int、1/x、squrt	常规算术函数
自定义函数	F1、F2、F3、F4	

2． 特殊函数键：

在 special function 的下拉框中有下列函数，如表所示。

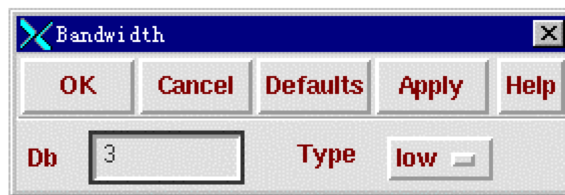
函数名	说明
Ishift	X 轴位移
Clip	在 clip 函数限制的范围内画波形
convolution	取两个波形的卷积

Eex	指数函数
Frequency	估计周期（准周期）波形的周期
GainBWprod	增益带宽积
Gain Margin	增益裕量
Phase Margin	相位裕量
Rise Time	上升时间
Slew Rate	摆率
bandwidth	带宽

下面将举例说明计算器波形处理功能的应用。如已得到如图 1-8-2 的电压的交流响应波形图，要计算它的-3dB 带宽。

步骤如下：

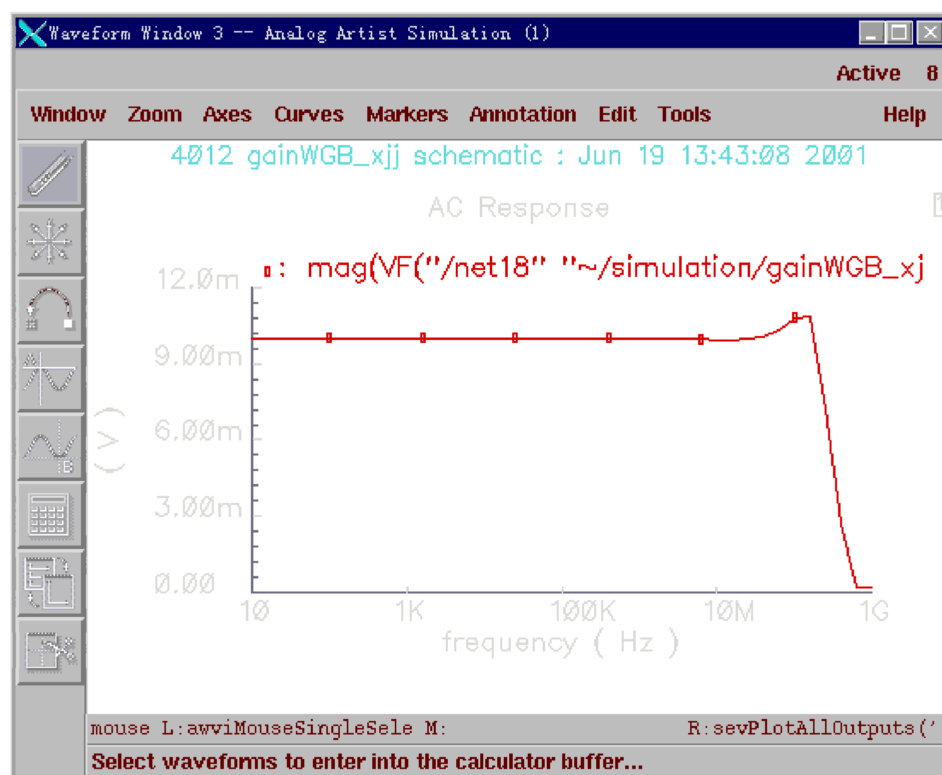
1）点击左边的 wave 键，然后在波形图中点击波形，在计算器的显示窗口中就会显示出该波形的名称；
2）在 special function 的下拉框中选择 bandwidth，得到如下窗口，在 Db 处填 3，在 Type 处选择 low（表示低通，high 表示高通，band 表示带通）,然后 ok。



3）点击 erplot 键，就可以在 waveform 窗口得到结果如图 1-8-3 所示。

处理波形：

4）点击左边的 wave 键，然后在电路原理图中选中所需要的波形，拖至计算器的命令行处，此处就会显示该波形的名称；
5）再结合右边的函数键，得到想要的表达式。如要得到分贝的形式，就点击 dB10 或 dB20 的键。
6）点击左边的 plot 键，就可以在 waveform 窗口得到结果。



1-8-2　　交流响应波形图



图 1-8-3　db 表示图

第二章． Virtuoso Editing 的使用简介

全文将用一个贯穿始终的例子来说明如何绘制版图。这个例子绘制的是一个最简单的非门的版图。

§2－1　建立版图文件

使用 library manager。首先，建立一个新的库 myLib，关于建立库的步骤，在前文介绍 cdsSpice 时已经说得很清楚了，就不再赘述。与前面有些不同的地方是：由于我们要建立的是一个版图文件，因此我们在 technology file 选项中必须选择 compile a new tech file,或是 attach to an exsiting tech file。这里由于我们要新建一个 tech file，因此选择前者。这时会弹出 load tech file 的对话框，如图 2-1-1 所示。

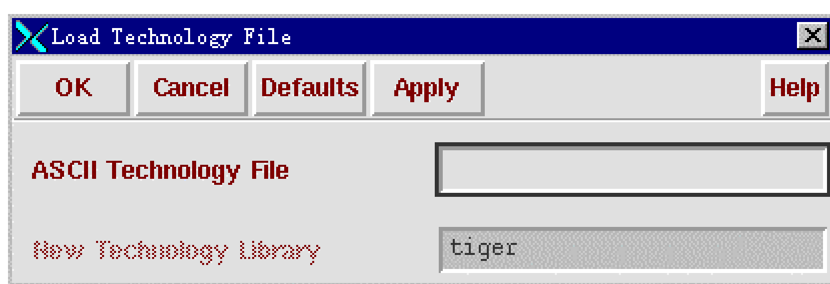


图 2-1-1

在 ASCII Technology File 中填入 csmc1o0.tf 即可。接着就可以建立名为 inv 的 cell 了。为了完备起见，读者可以先建立 inv 的 schematic view 和 symbol view（具体步骤前面已经介绍，其中 pmos 长 6u，宽为 0.6u。nmos 长为 3u，宽为 0.6u。model 仍然选择 hj3p 和 hj3n)。然后建立其 layout view，其步骤为：在 tool 中选择 virtuoso－layout，然后点击 ok。

§2－2 绘制 inverter 掩膜版图的一些准备工作

首先，在 library manager 中打开 inv 这个 cell 的 layout view。即打开了 virtuoso editing 窗

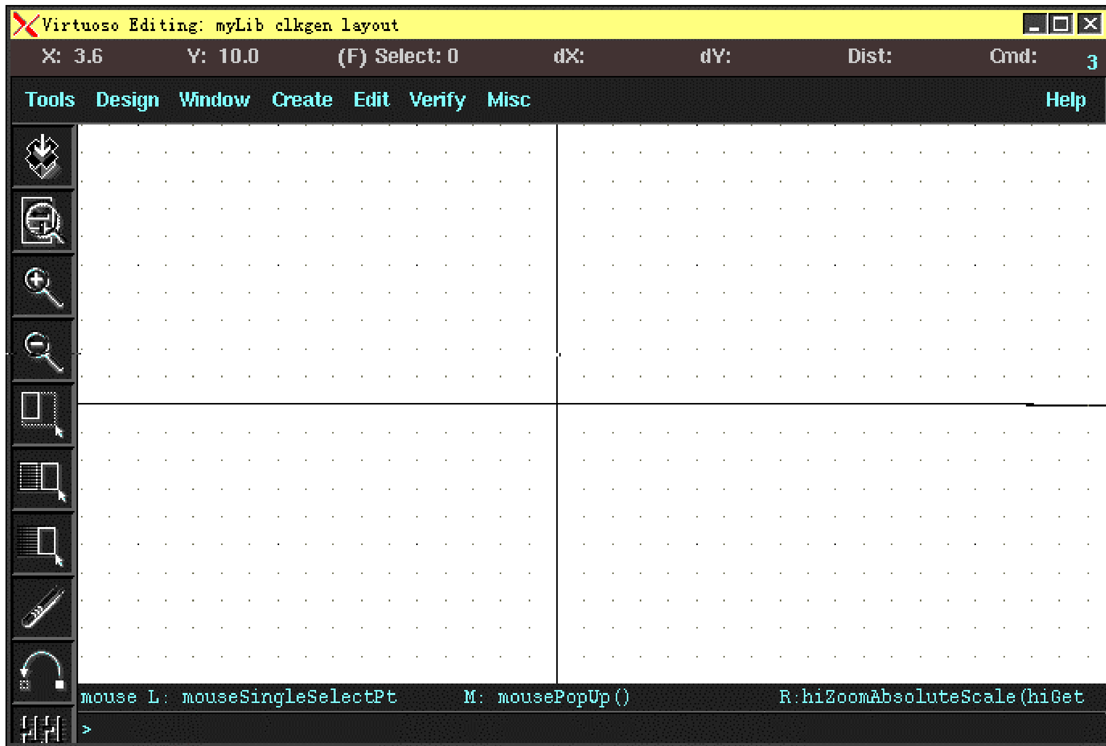


图 2-2-1　virtuoso editing 窗口

口，如图 2-2-1 所示。

版图视窗打开后，掩模版图窗口显现。视窗由三部分组成：**Icon menu , menu banner , status banner.**

Icon menu (图标菜单)缺省时位于版图图框的左边，列出了一些最常用的命令的图标，要查看图标所代表的指令，只需要将鼠标滑动到想要查看的图标上，图标下方即会显示出相应的指令。

menu banner（菜单栏）,包含了编辑版图所需要的各项指令，并按相应的类别分组。几个常用的指令及相应的快捷键列举如下：

Zoom In -------放大 (z)　　**Zoom out by 2**------- 缩小 2 倍(Z)

Save ------- 保存编辑(f2)　　**Delete** ------- 删除编辑(Del)

Undo ------- 取消编辑(u)　　**Redo** -------恢复编辑 (U)

Move ------- 移动(m)　　**Stretch** ------- 伸缩(s)

Rectangle -------编辑矩形图形(r)　　**Polygon** ------- 编辑多边形图形(P)

Path ------- 编辑布线路径(p)　　**Copy** -------复制编辑 (c)

status banner（状态显示栏），位于 **menu banner** 的上方，显示的是坐标、当前编辑指令等状态信息。

在版图视窗外的左侧还有一个层选择窗口（**Layer and Selection Window　LSW**）。

LSW 视图的功能：

1） 可选择所编辑图形所在的层；

2） 可选择哪些层可供编辑；

3） 可选择哪些层可以看到。

由于我们所需的部分版图层次在初始 LSW 中并不存在，因此下一步要做的是：建立我们自己的工艺库所需的版图层次及其显示属性。为了简单起见，以下仅列出绘制我们这个版图所需的最少版图层次。

层次名称	说明
Nwell	N 阱
Active	有源区
Pselect	P 型注入掩膜
Nselect	N 型注入掩膜
Contact	引线孔，连接金属与多晶硅/有源区
Metal1	第一层金属，用于水平布线，如电源和地
Via	通孔，连接 metal1 和 metal2
Metal2	第二层金属，用于垂直布线，如信号源的 I/O 口
Text	标签
Poly	多晶硅，做 mos 的栅

下图是修改后的 LSW。

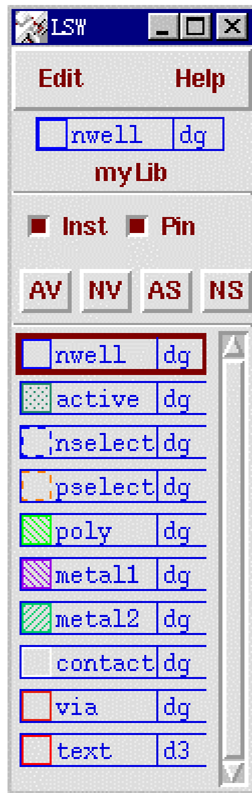


图 2-2-2 LSW

如何来修改 LSW 中的层次呢？以下就是步骤：

1． 切换至 CIW 窗口，在 technology file 的下拉菜单中选择最后一项 edit layers 出现如图窗口

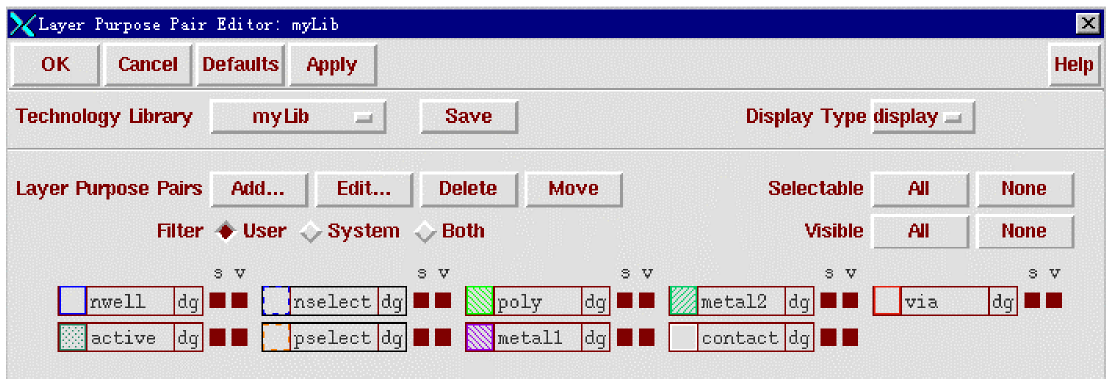


图 2-2-3 edit layers

2． 在 technology library 中选择库 mylib，先使用 delete 功能去除不需要的层次。然后点击 add 添加必需的层次，add 打开如下图的窗口：

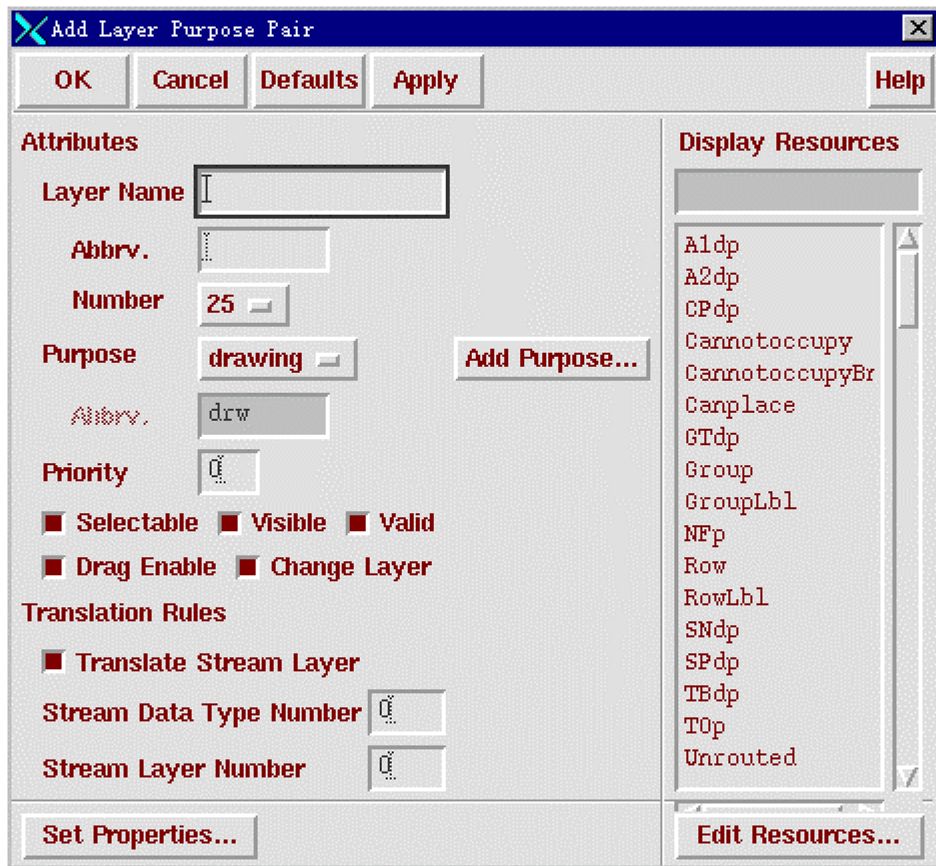


图 2-2-4

其中，layer name 中填入所需添加的层的名称。Abbv 是层次名称缩写。Number 是系统给层次的内部编号，系统保留 128－256 的数字作为其默认层次的编号而将 1－127 留给开发者创造新层次。Purpose 是所添加层次的功用，如果是绘图层次，一般选择 drawing。Priority 是层次在 LSW 中的排序位置。其余的选项一般保持默认值。在右边是图层的显示属性。可以直接套用其中某些层次的显示属性。也可以点击 edit resources 自己编辑显示属性。如图 2-2-5 所示（这个窗口还可以在 LSW 中调出） 编辑方法很简单，读者可以自己推敲，就不再赘述。上述工作完毕后就得到我们所需的层次。接着我们就可以开始绘制版图了。

§ 2—3　绘制版图

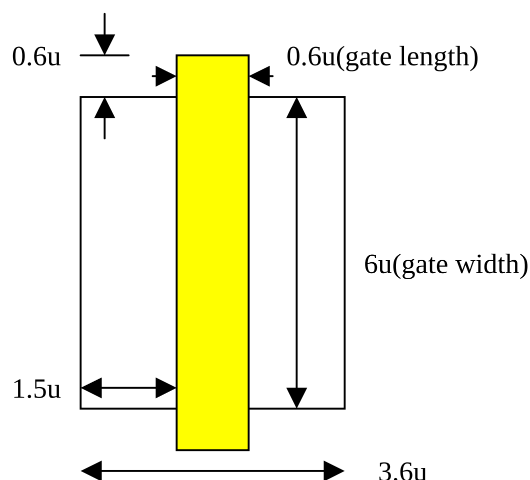
一．画 pmos 的版图（新建一个名为 pmos 的 cell）

1．画出有源区

在 LSW 中，点击 active（dg），注意这时 LSW 顶部显示 active 字样，说明 active 层为当前所选层次。然后点击 icon menu 中的 rectangle icon，在 vituoso editing 窗口中画一个宽为 3.6u，长为 6u 的矩形。这里我们为了定标，必须得用到标尺。点击 misc/ruler 即可得到。清除标尺点击 misc/clear ruler。如果你在绘制时出错，点击需要去除的部分，然后点击 delete icon。

2．画栅

在 LSW 中，点击 poly（dg），画矩形。与有源区的位置关系如下图：



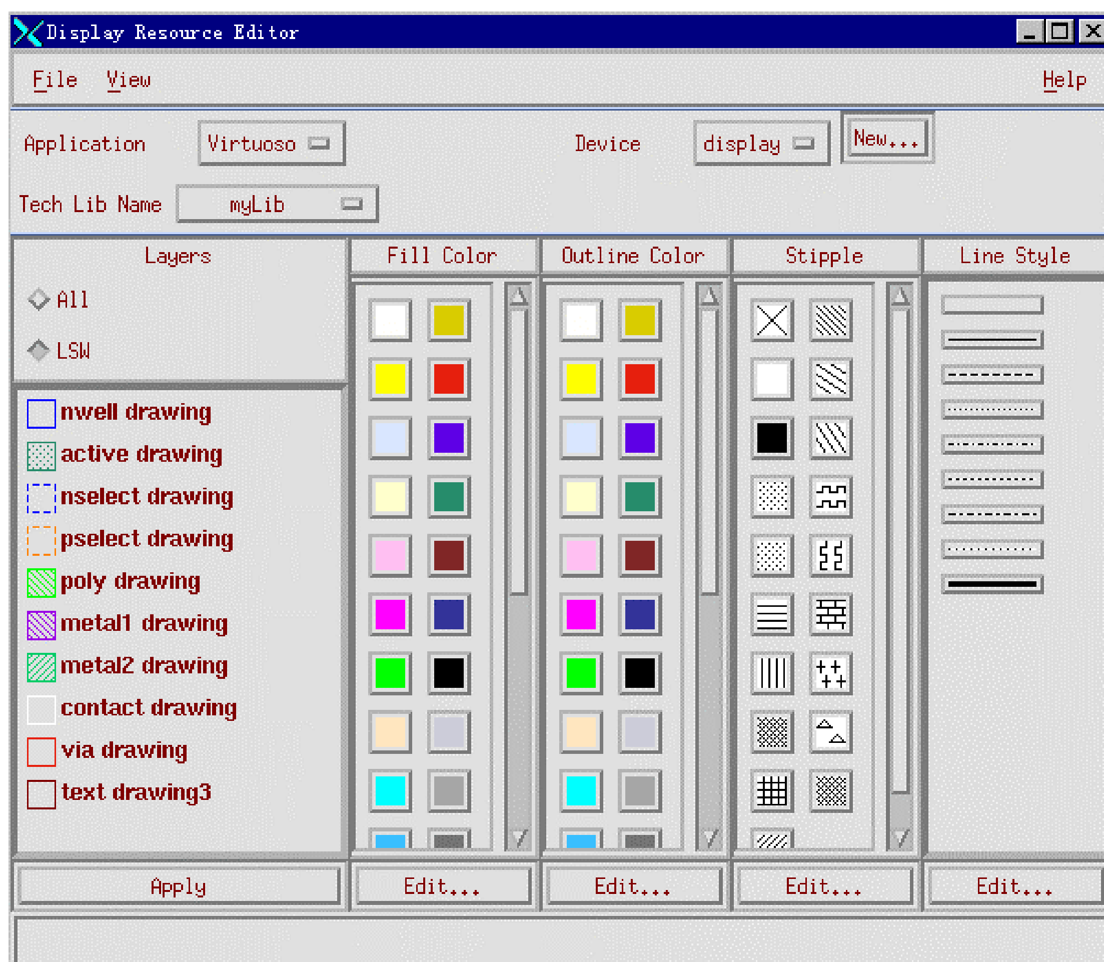
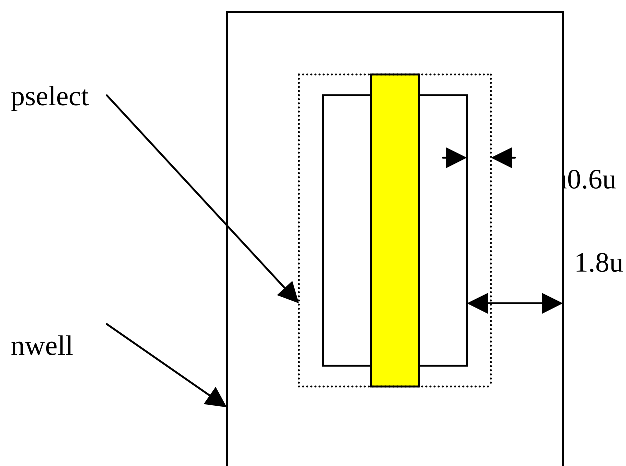


图 2-2-5　display resource editor

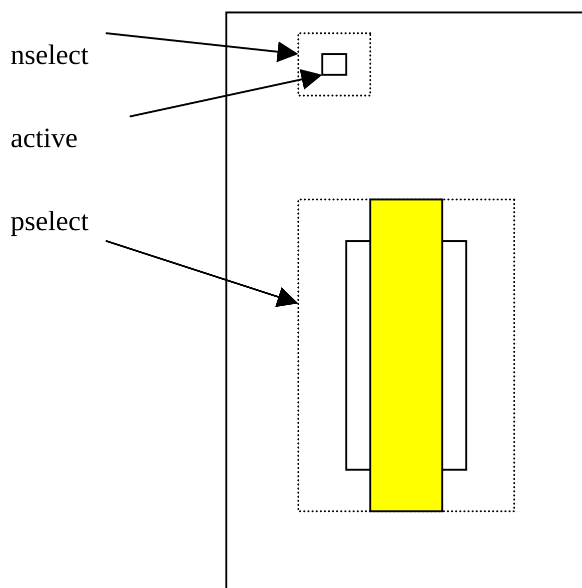
3．画整个 pmos

为了表明我们画的是 pmos 管，我们必须在刚才图形的基础上添加一个 pselect 层，这一层将覆盖整个有源区 0.6u。接着，我们还要在整个管子外围画上 nwell，它覆盖有源区 1.8u。如下图所示：



4．衬底连接

pmos 的衬底（nwell）必须连接到 vdd。首先，画一个 1.2u 乘 1.2u 的 active 矩形；然后在这个矩形的边上包围一层 nselect 层（覆盖 active0。6u）。最后将 nwell 的矩形拉长，完成后如下图所示：



这样一个 pmos 的版图就大致完成了。接着我们要给这个管子布线。

二．布线

pmos 管必须连接到输入信号源和电源上，因此我们必须在原图基础上布金属线。

1．首先我们要完成有源区（源区和漏区）的连接。在源区和漏区上用 contact（dg）层分别画三个矩形，尺寸为 0.6 乘 0.6。注意：contact 间距为 1.5u。
2．用 metal1（dg）层画两个矩形，他们分别覆盖源区和漏区上的 contact，覆盖长度为 0.3u。
3．为完成衬底连接，我们必须在衬底的有源区中间添加一个 contact。这个 contact 每边都被 active 覆盖 0.3u。
4．画用于电源的金属连线，宽度为 3u。将其放置在 pmos 版图的最上方。

布线完毕后的版图如下图所示：

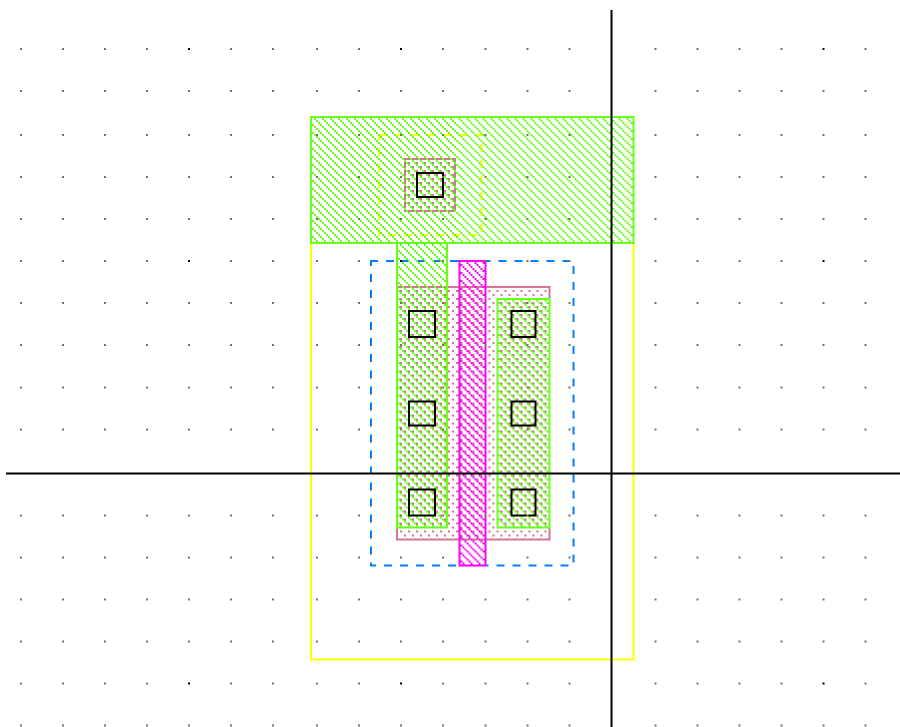


图 2-3-1　pmos 版图

通过以上步骤我们完成了 pmos 的版图绘制。接下来我们将绘制出 nmos 的版图。

三．画 nmos 的版图

绘制 nmos 管的步骤同 pmos 管基本相同（新建一个名为 nmos 的 cell）。无非是某些参数变化一下。下面给出 nmos 管的图形及一些参数，具体绘制步骤就不再赘述。

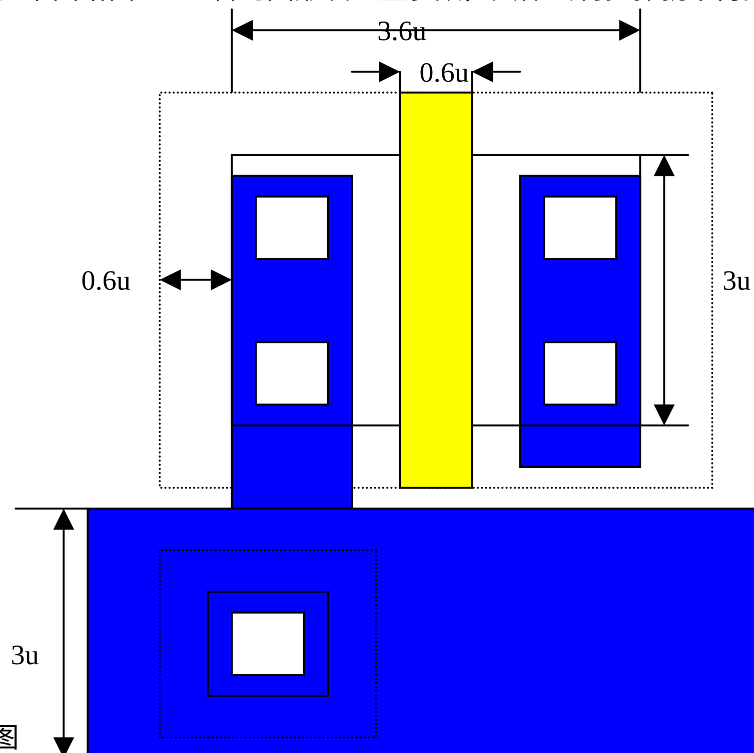


图 2-3-2nmos 版图

四．完成整个非门的绘制及绘制输入、输出

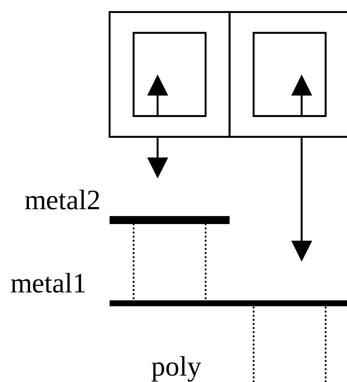
1．新建一个 cell（inv）。将上面完成的两个版图拷贝到其中，并以多晶硅为基准将两

图对齐。然后，我们可以将任意一个版图的多晶硅延长和另外一个的多晶硅相交。

2． 输入：为了与外部电路连接，我们需要用到 metal2。但 poly 和 metal2 不能直接相连，因此我们必须得借助 metal1 完成连接。具体步骤是：

   a． 在两 mos 管之间画一个 0.6 乘 0.6 的 contact
   b． 在这个 contact 上覆盖 poly，过覆盖 0.3u
   c． 在这个 contact 的左边画一个 0.6 乘 0.6 的 via，然后在其上覆盖 metal2（dg），过覆盖 0.3u
   d． 用 metal1 连接 via 和 contact，过覆盖为 0.3u

从下图中可以看得更清楚：



3． 输出：先将两版图右边的 metal1 连起来（任意延长一个的 metal1，与另一个相交）。然后在其上放置一个 via，接着在 via 上放置 metal2。

五．作标签

1． 在 LSW 中选择层次 text（d3），点击 create/label，在弹出窗口中的 label name 中填入 vdd！并将它放置在版图中相应的位置上。

2． 按同样的方法创制 gnd！、A 和 Out 的标签。完成后整个的版图如下：

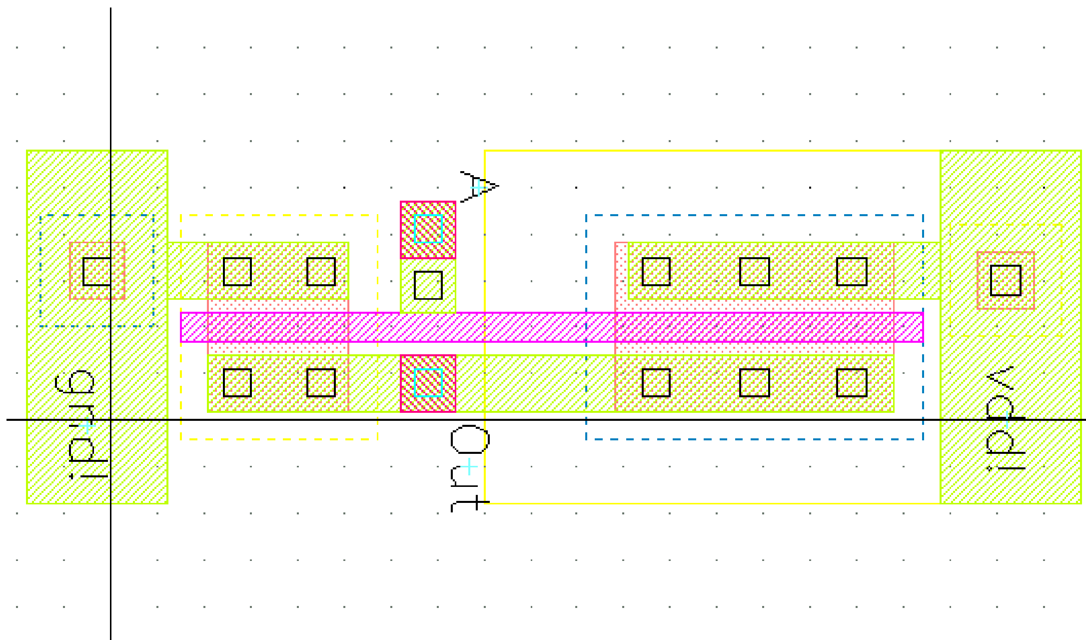


图 2-3-4 非门的版图

至此，我们已经完成了整个非门的版图的绘制。下一步将进行 DRC 检查，以检查版图在绘制时是否有同设计规则不符的地方。

第三章 Diva 验证工具使用说明

版图绘制要根据一定的设计规则来进行，也就是说一定要通过 DRC（Design Rule Checker）检查。编辑好的版图通过了设计规则的检查后，有可能还有错误，这些错误不是由于违反了设计规则，而是可能与实际线路图不一致造成。版图中少连了一根铝线这样的小毛病对整个芯片来说都是致命的，所以编辑好的版图还要通过 LVS（Layout Versus Schematic）验证。同时，编辑好的版图通过寄生参数提取程序来提取出电路的寄生参数，电路仿真程序可以调用这个数据来进行后模拟。下面的框图可以更好的理解这个流程。

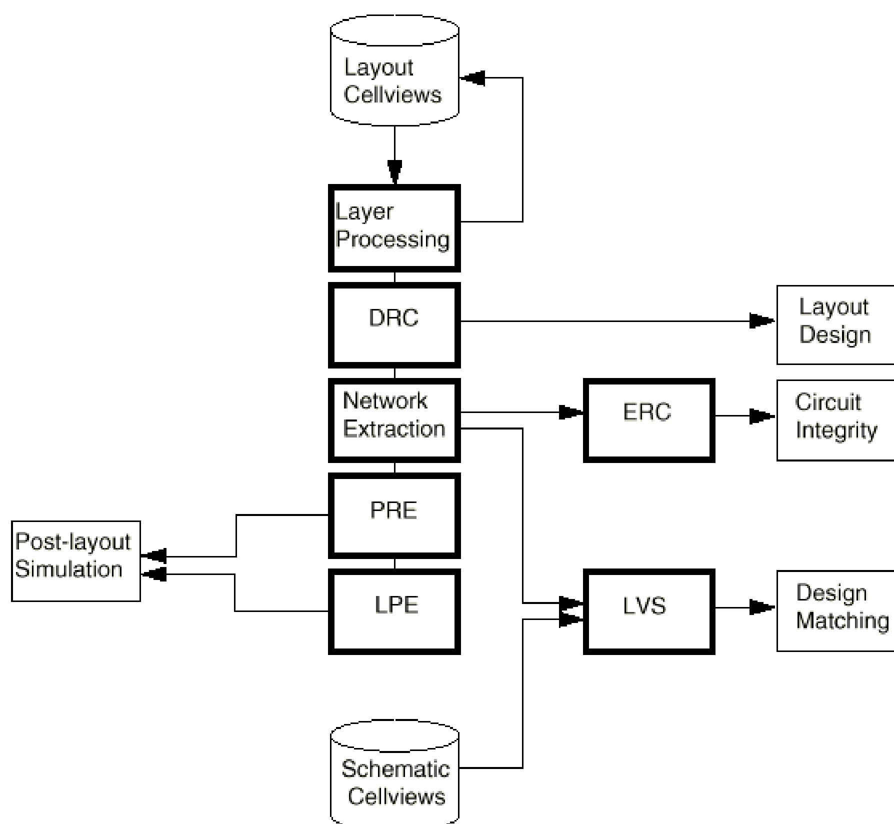


图 3-0-1 IC 后端工作流程

验证工具有很多，我们采用的是 Cadence 环境下集成的验证工具集 DIVA。下面先对 DIVA 作一个简单介绍。

DIVA 是 Cadence 软件中的验证工具集，用它可以找出并纠正设计中的错误：它除了可以处理物理版图和准备好的电气数据，从而进行版图和线路图的对查（LVS）外。还可以在设计的初期就进行版图检查，尽早发现错误并互动地把错误显示出来，有利于及时发现错误所在，易于纠正。

DIVA 工具集包括以下部分：

1．设计规则检查（iDRC）
2．版图寄生参数提取（iLPE）
3．寄生电阻提取（iPRE）

4．电气规则检查（iERC）
5．版图与线路图比较程序（iLVS）

需要提到的是：Diva 中各个组件之间是互相联系的，有时候一个组件的执行要依赖另一个组件先执行。例如：要执行 LVS 就先要执行 DRC。在 Cadence 系统中，Diva 集成在版图编辑程序 Virtuoso 和线路图编辑程序 Composer 中，在这两各环境中都可以激活 Diva。要运行 Diva 前，还要准备好规则验证的文件。可以把这个文件放在任何目录下，这些规则文件的写法下面专门会进行说明，也会给出例子。这些文件有各自的默认名称，如：做 DRC 时的文件应以 divaDRC.rul 命名，版图提取文件以 divaEXT.rul 命名。做 LVS 时规则文件应以 divaLVS.rul 命名。

§3－1 DRC 规则文件的编写

仍旧以前面的非门为例，我们制定了以下规则：

1.a	n 阱(well)	n 阱的最小宽度	4.8u
1.b		阱与阱之间的最小间距	1.8u
1.c		ndiff 到 nwell 的最小间距	0.6u
1.d		pdiff 到 nwell 的最小间距	1.8u
1.e		p mos 器件必须在 nwell 内	
2.a	有源区（active）	有源区的最小宽度	1.2u
2.b		有源区之间的最小间距	1.2u
3.a	多晶硅（poly）	多晶硅的最小宽度	0.6u
3.b		多晶硅间的最小宽度	0.6u
3.c		多晶硅与有源区的最小间距	0.6u
3.d		多晶硅栅在场区上的最小露头	0.6u
3.e		源、漏与栅的最小间距	0.6u
4.a	引线孔（contact）	引线孔的最小宽度	0.6u
4.b		引线孔间的最小间距	0.9u
4.c		多晶硅覆盖引线孔的最小间距	0.3u
4.d		metal1 覆盖引线孔的最小间距	0.3u
5.a	金属 1（metal1）	金属 1 的最小宽度	1.2u
5.b		金属 1 间的最小间距	0.9u
6.a	金属 2（metal2）	金属 2 的最小宽度	1.2u
6.b		金属 2 间的最小间距	1.2u
6.c		金属 2 的最小挖槽深度	1.2u
7.a	通孔（via）	通孔的最小宽度	0.6u
7.b		通孔间的最小间距	0.9u
7.c		通孔与引线孔间的最小间距	0.6u
7.d		metal1 覆盖通孔的最小间距	0.3u

7.e	metal2 覆盖通孔的最小间距	0.3u
7.f	通孔与多晶硅的最小间距	0.3u

结合上述规则，我们就可以编写出相应的 DRC 规则检查文件（见附录 1），取名为 divaDRC.rul。这个文件的第一部分是层次处理，用于生成规则文件中所要应用到的层次（可以是原始层或是衍生层）。例如：nwell=geomOr("nwell")，(在文件中引用到的所有原始物理层次都要用双引号括起来）这一句的目的是在后面应用到 nwell 这个原始物理层次时，不需要再用引号括起来，前面几句都是这个意思。后面四句则生成版图验证中必须的一些层次。有一点需要注意的是：在 geomOr 的关键字和“ (”之间不能出现空格，nwell=geomOr (“nwell”)的写法系统在编译时会报错。

下面这个语句相当于一个条件转移语句，当有drc命令时，执行下面的规则，否则跳转到下一个命令。

```
ivIf( switch( "drc?" ) then
```

在设计规则检查中，主要的语句就是drc（）了。先简单介绍一下这个语句的语法。

```
[outlayer]=drc(inlayer1 [inlayer2] function [modifiers] )
```

outlayer表示输出层，如果定义（给出）输出层，则通过drc检查的出错图形就可以保存在该输出层中。此时，如果没有modifiers选项，则保存的是原始的图形。如果在modifiers选项中定义了修改方式，那么就把修改后的结果保存在输出层中。如果没有定义outlayer层，出错的信息将直接显示在出错的原来层次上。

Inlayer1和inlayer2代表要处理的版图层次。有些规则规定的是只对单一层次的要求，比如接触孔的宽度，那么可以只有inlayer1。而有些规则定义的是两个层次之间的关系，如接触孔和铝线的距离，那么要注明两个层次。

Function中定义的是实际检查的规则，关键字有sep（不同图形之间的间距），width（图形的宽度），enc（露头），ovlp(过覆盖)，area（图形面积），notch（挖槽的宽度）等。关系有>，<，>=，<=，==等。结合起来就是：sep<3，width<4，1<enc<5 这些关系式。例如：drc(nwell width < 4.8 "Minimum nwell width =4.8")。在此例中，没有outlayer的定义，也没有modifiers的定义，所以发现的错误都直接显示在nwell层上。例子中，inlayer就是nwell，检查的只是n阱层的规则。function是width<4.8，表示n阱宽度小于4.8微米。所以上面这句的执行结果就是把n阱层中宽度小于4.8u的图形当做错误输出。后面引号中的信息起到说明提示作用，需要时可以查询，对查错没有实际意义。同样需要注意的是：在drc和“ （ ”之间同样不能有空格，否则系统会提示没有drc语句。从上面讨论不难看出，DIVA规则文件的编写对格式有一定要求。

在规则文件中我们还可以看到saveDerived语句，如：saveDerived(geomAndNot(pgate nwell) "p mos device must in nwell")，这一句将输出不在nwell内部的pgate（pmos），这种写法在规则文件的编写中经常碰到，要熟练掌握。

另外，在DRC文件中，引号引出的行是注释行。

以上就是对DRC文件编写的一些简单介绍，对于其中使用的关键字，作者有专门的说明文章，同时在本文后面作者还会给出一个完整的DRC校检文件并给出详细说明，读者可以参照它，以加深对文件编写的理解。

§3－2 版图提取文件的介绍

上面已经提到，通过DRC验证的版图还需要进行LVS也就是版图和线路图对查比较。实际上就是从版图中提取出电路的网表来，再与线路图的网表比较。那么如何提取版图网表呢？这里我们就要使用到DIVA的extract文件。下面是它的简单介绍：

首先，同DRC一样，extract文件的最开始同样是这样一条语句：

ivIf（switch（“extract？”）　then
它相当于一个条件转移语句，当有extract这个命令时，执行下面的规则，否则跳转到另外的循环。

接着，extract文件中要进行的是层次定义，它一般分为三个步骤：

1．识别层定义（recognition layer）
2．终端层定义（terminal layer）
3．伪接触层定义（psuedo_contact layer）

然后是定义层次间的连接关系，使用geomConnect语句将版图间的不同层次连接起来（一个extract文件只能有一个geomConnect语句），构成完整的网表。例如句子：

```
geomConnect (
via (contact psd nsd poly metal1)
via (via metal1 metal2)
)
```

其中，via语句的作用是使用连接层连接任意数目的层次，但要注意的是：一个via语句中只能出现一个连接层。但在geomConnect语句中via语句可以出现的次数不限。以上语句表示：在有contact的地方，psd nsd poly metal1 是相互连接的。在有via 的地方metal1和metal2相连，注意后一个via和前一个的意义不同。

上述工作完成之后，我们接着要进行的工作是器件的提取（device extraction）。使用extractDevice语句。extractDevice 语句定义电路中用到的元器件，这是提取文件中的关键语句。语法说明如下：

```
extractDevice( reclayer termlayer model physical )
```

其中reclayer是识别层，它应该是后来通过逻辑关系生成的提取层，这个层上的每一个图形都会被当作是一个元器件。

Termlayer是端口层，它表示的是元器件的端口，一定要是可以连接的层次。具体的端口定义因元器件而异。

Model指的是元器件的类型，与端口要对应。例如下两句：

```
extractDevice( pgate (GT "G")(psd "S" "D")(NT "B")"pfet ivpcell" )
extractDevice( ngate (GT "G")(nsd "S" "D")(pwell "B")"nfet ivpcell" )
```

分别提取出pmos管和nmos管。

接着很重要的一步是器件尺寸测量，使用measureParameter语句，例如：

w1＝measureParameter（length （ngate butting nsd） .5）

这一句测量的是nmos的沟道宽度,注意后面的.5必须加上,否则测出的将是两倍的沟道宽度。

下面使用saveInterconnect 这个命令把连接的层次写到提取出来的网表中，以便在做LVS时，可以与线路图中的网表互相对比。

```
saveInterconnect( nsd psd poly contact metal1 )
```

saveRecognition 这个命令将提取产生的可以识别的图形保存下来。通常和extractDevice语句中的识别层一致。

```
saveRecognition( ngate "ngate" )
saveRecognition( pgate "pgate" )
```

以上就是对extract文件的一个简要介绍，读者可以参看附录中完整的例子，以加深对它的理解。

§3－3　LVS文件的介绍

接下来，就是LVS检查了。在diva中，由于版图提取在extract中就已经完成，LVS文件

中的逻辑结构相对就比较简单。只需进行网表比较，参数比较，以及把一些“并联或串联”的元器件归并等即可。所以这一部分文件不会因为工艺层次不同而有很大不同，可以根据范本做少许改动。

以下只介绍一下LVS的基本结构：

```
  lvsRules (
   procedure(mosCombine(value1,value2)
      …….
  )
   Procedure(mosCompare(lay,sch)
     …….
)
permuteDevice(parallel “pmos” mosCombine)
compareDeviceProperty(“pmos” mosCompare)
)
```

至于例子，读者可以参考附录。

§3—4 Diva 的用法

一．DRC 的说明

编辑好的验证文件都存在..\export\home\wmy\myLib\下，文件名分别是divaDRC.rul、divaEXT.rul、divaLVS.rul。有了这三个文件就可以进行版图验证了。下面将以一个非门为例子来进行说明。

在编辑版图文件的同时就可以进行DRC检查。在virtuoso版图编辑环境中。单击Verify菜单，上面提到的DIVA工具都集成在这个菜单下。先介绍设计规则检查DRC，单击第一个子菜单DRC就会弹出DRC的对话框。如下：

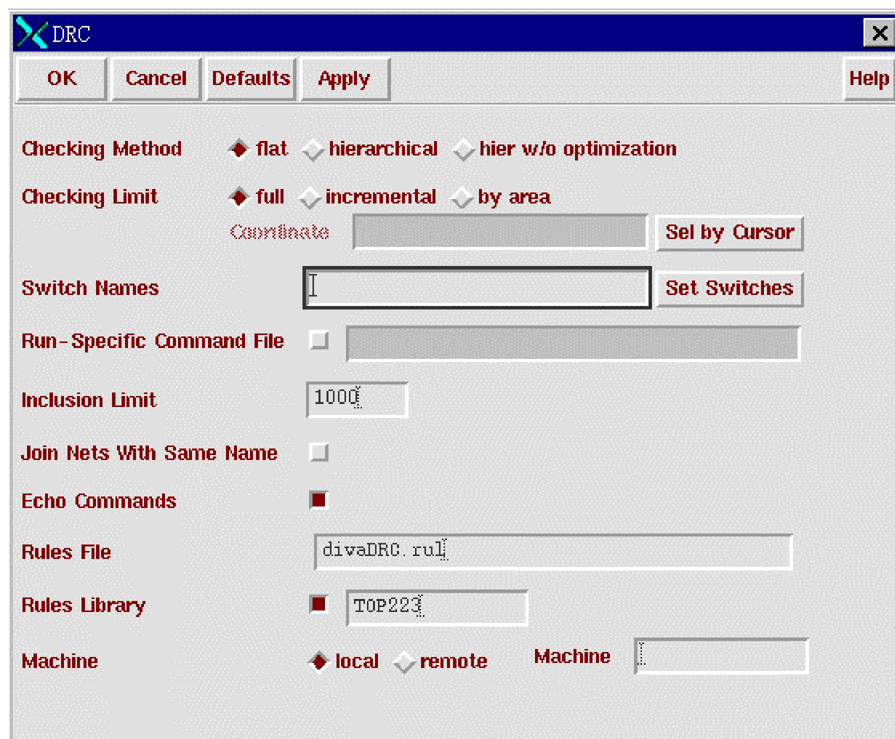


图 3-4-1 DRC 菜单窗口

Checking Method 指的是要检查的版图的类型。

Flat 表示检查版图中所有的图形，对子版图块不检查。（与电路图中类似，最上层电路由模块组成，而模块由小电路构成。有些复杂的版图也是如此）

Hierarchical 利用层次之间的结构关系和模式识别优化，检查电路中每个单元块内部是否正确。

hier w/o optimization 利用层次之间的结构关系而不用模式识别优化，来检查电路中每个单元块。

Checking Limit 可以选择检查哪一部分的版图

Full 表示查整个版图

Incremental 查自从上一次 DRC 检查以来，改变的版图。

by area 是指在指定区域进行 DRC 检查。一般版图较大时，可以分块检查。

如果选择这种方式后，*Coordinate* 这个输入框就变为可输入。可以在这个框内输入坐标，用矩形的左下角和右上角的坐标来表示。格式为：`12599:98991 115682:194485`
或者先单击 `Sel by Cursor`,然后用鼠标在版图上选中一个矩形，这个输入框也会出现相应的坐标。如果不出现可以多选几次。

Switch Names
在DRC文件中，我们设置的switch在这里都会出现。这个选项可以方便我们对版图文件进行分类检查。这在大规模的电路检查中非常重要。

Run-Specific Command File

Inclusion Limit

上面的两项并不是必需的，可以根据默认设定。

Echo Commands 选上时在执行DRC的同时在CIW窗口中显示DRC文件。

Rules File 指明DRC规则文件的名称，默认为divaDRC.rul

Rules Library 这里选定规则文件在哪个库里。

Machine 指明在哪台机器上运行DRC命令。

local 表示在本机上运行。对于我们来说，是在本机运行的，选local。

remote 表示在远程机器上运行。

Remote Machine Name 远程机器的名字。

在填好规则文件的库和文件名后，根据实际情况填好**Checking Method** 和**Checking Limit**就可以单击OK运行。这时可以在CIW窗口看到运行的信息，同时在版图上也会出现发亮的区域（如果有错误）。

错误在版图文件中可以看到，另外也可以选择***Verify-Markers-Find***菜单来帮助找错。单击菜单后会弹出一个窗口，在这个窗口中单击***apply***就可以显示第一个错误。这个窗口较简单，大家看一下，再试几次就可以了。

同样，可以选择***Verify-Markers-Explain***来看错误的原因提示。选中该菜单后，用鼠标在版图上出错了的地方单击就可以了。也可以选择***Verify-Markers-Delete***把这些错误提示删除。

Virtuoso版图编辑环境下的菜单见图3-4-2。

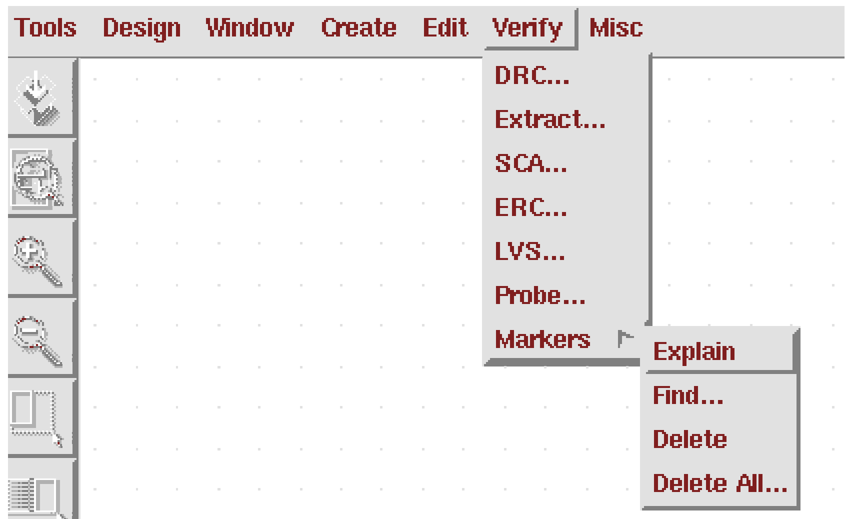


图 3 –4-2Virtuoso 菜单

二．版图提取（Extractor）说明

为了进行版图提取，还要给版图文件标上端口，这是LVS的一个比较的开始点。在LSW窗口中，选中metal1（pn）层，（pn）指得是引脚（pin）；然后在Virtuoso环境菜单中选择***Create-Pin***，这时会出来一个窗口。如下：

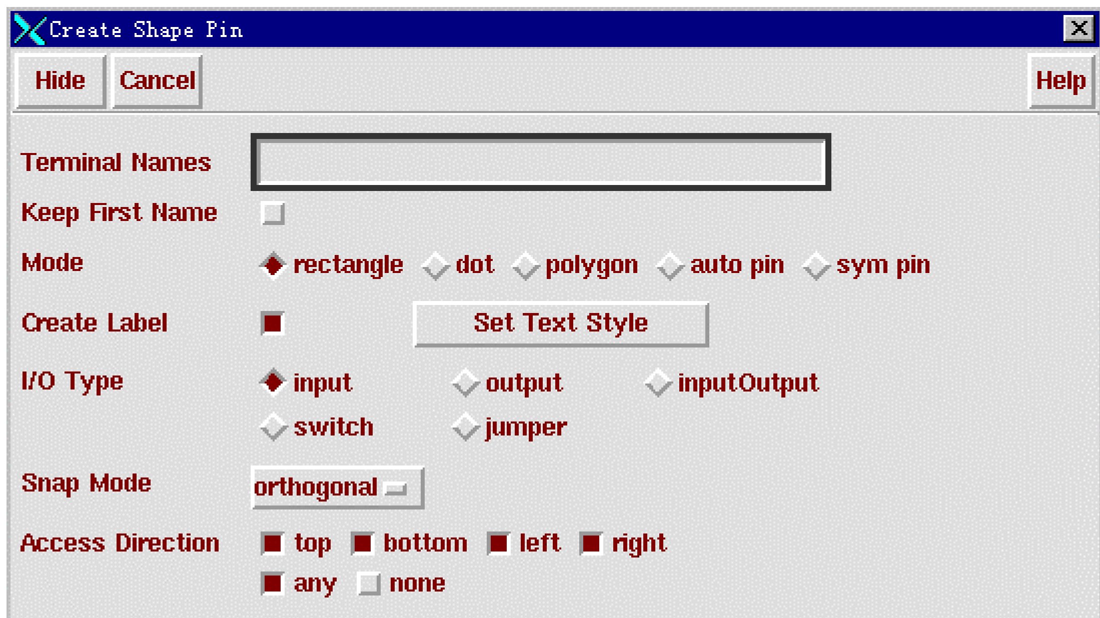


图 3-2-3 创建版图端口窗口

填上端口的名称（**Terminal Names** 和Schematic中的名字一样）、模式（**Mode**，一般选rectangle）、输入输出类型（**I/O Type**）等。至于**Create Label**属于可选择项，选上后，端口的名称可以在版图中显示。

填好可以直接在版图中画上端口，往往有好几个端口，可以都画好在单击Hide。 这些端口仅表示连接关系，并不生成加工用的掩模板，只要求与实际版图上铝线接触即可，也没有规则可言。

版图的完成后，就可以提取了，在版图编辑环境下选择***Verify –extractor*** 。弹出菜单如下：

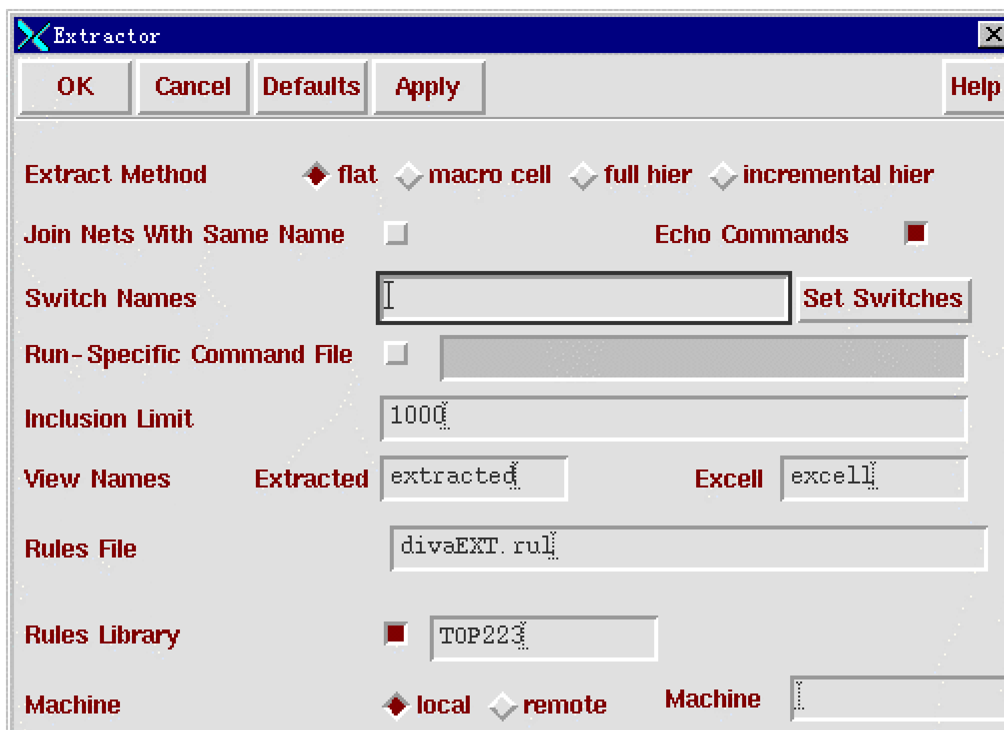


图 3-2-4Extractor 窗口

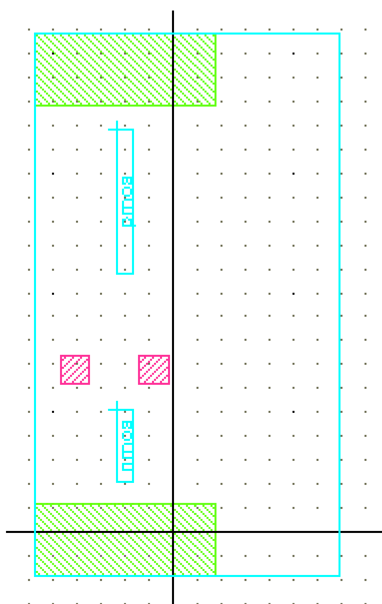


图 3-2-5 提取出的文件

填好提取文件库和文件名后，单击OK就可以了。然后打开Library Manager，在库myLib下nmos单元中增加了一个文件类型叫extracted的文件，可以用打开版图文件同样的方式打开它。图3-2-5就是提取出来的版图，可以看到提取出来的器件和端口，要看连接关系的话，可以选择***Verify-probe***菜单，在弹出窗口中选择查看连接关系。

版图的准备工作基本上就完成了，接下来是线路图的准备工作。线路图的准备工作相

对较简单，有几个要注意的地方：首先，在库的选用上，要用Sample库中的元件；其次，线路图的端口名称要与版图中的端口名称一致；最后，在线路编辑完成后要进行检查，可以直接单击左边第一个快捷键，也可以选择菜单***Check--Current Cellview***。

在版图和线路图的准备工作完成后就可以进行LVS了。

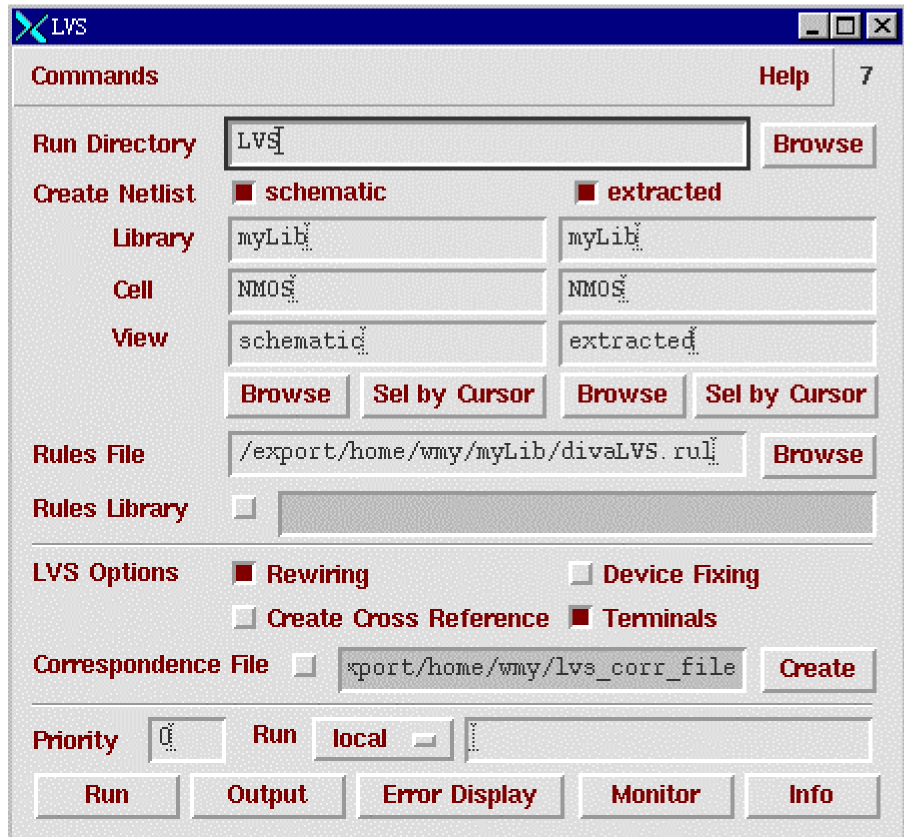


图3-2-6　LVS

参照图3-2-6的弹出菜单，填好规则文件的库和文件名，要进行LVS的两个网表。（其实在LVS中比较的是两个网表，一个是schematic中，另一个是extracted，所以两个schematic文件也可以比较，只是一般没这个必要）设置完以后单击RUN，片刻后就回弹出一个窗口表示LVS完成或者失败。失败时可以在上面的菜单中单击Info看运行的信息再进行处理。LVS完成后，可以在上面的弹出菜单中单击Output，这时会弹出LVS的结果。

当然，LVS完成并不是说LVS通过了，可能会有很多地方不匹配。这时要查看错误可以在LVS窗口中单击Error Display。即可在Extracted和Schematic 中查看错误。

第四章 Cadence 中 Verilog 的一些使用方法

§4－1 Verilog 的文本编辑器

随着电路规模的增大和复杂，传统的图形输入模式已不可行。语言描述电路成为潮流。它的方便性和好的更改性、维护性在实践中得到很好的体现。尤其现在强大的综合工具，和系统集成对核的需求性使 Verilog 更有用武之地。每个硬件工程师应该学习掌握它。

在进入 **Cadence** 后在命令行中键入

textedit *.v↙

(此处*为文件名，在 textedit 命令后应带上文件名)

键入上述命令后进入文本编辑框，和 Windows 中常用的文本编辑框很象。

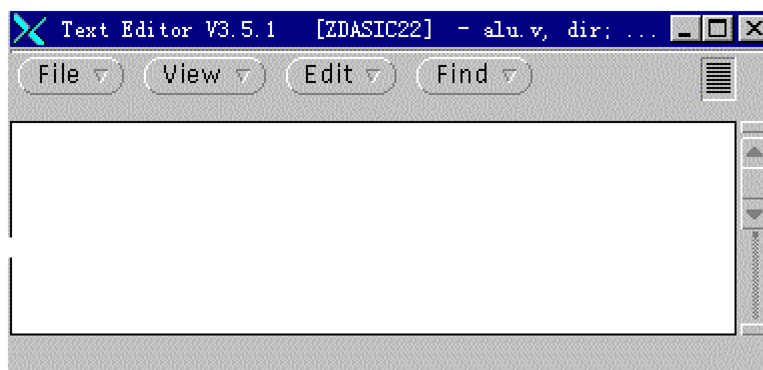


图 4-1-1textedit 文本编辑框界面

图中的主菜单 File、View、Edit、Find 及各自底下的子菜单和 Windws 中的文本编辑器差不多，使用方法相似，这里就不多说了。编好程序保存可以进行后续工作了。

§4－2 Verilog 的模拟仿真

一．命令的选择。

在命令行中键入

verilog↙

会出现关于此命令的一些介绍，如下：

```
-f <filename>  read host command arguments from file.
-v <filename>  specify library file
-y <filename>  specify library directory
-c compile only
-s enter interactive mode immediately
```

```
-k <filename>  set key file name
-u             convert identifiers to upper case
-t             set full trace
-q             quiet
-d             decompile data structure
Special behavioral performance options (if licensed):
  +turbo         speed up behavioral simulation.
  +turbo+2       +turbo with second level optimizations.
  +turbo+3       +turbo+2 with third level optimizations.
  +listcounts     generate code for maintaining information for
$listcounts
  +no_turbo      don't use a VXL-TURBO license.
  +noxl          disable XL acceleration of gates in all modules
Special environment invocation options (if licensed):
  +gui           invoke the verilog graphical environment
```

在上面的参数选择中，简单介绍几个常用的:

(1)-c

首先应该保证所编程序的语法正确性。先进行语法的检查，选择参数- c 键入如下命令。

verilog –c *.v↙

根据 Cadence 的报告，查找错误信息的性质和位置，然后进入文本编辑器进行修改，再编译，这是个反复的过程，直到没有语法错误为止。

(2)-s

进入交互式的环境，人机交互运行和下面的参数联合使用。

(3)+gui &

verilog 仿真有命令和图形界面两种方式。图形界面友好和 windows 使用很象，很好掌握，一般都使用图形方式。“&”符号是后台操作的意思，不影响前台工作。如此时你可以在命令行输入其它的命令。

其它的命令参数选择比较复杂，这里就不介绍了，故我们这里常用的命令是：

verilog –s *.v +gui &↙　　　(*代表文件名)

进入图形交互界面。

$附：命令行输入

!!↙

是执行上一条命令，

命令行输入

!* ↙　(*代表字母)

是执行最近的以*开头的命令。

上述附注对命令输入速度提高有所帮助。

二．SimVision 图形环境。

SimVision 是 Verilog-XL 的图形环境。 主要有 SimControl、Navigator、Signal Flow Browswer、 Wactch Objects Window 、SimWave 等窗口。

(1)SimControl 窗口

此窗口是主要的仿真控制窗口，让用户和机器进行交互式操作。执行各种Verilog-XL 命令(菜单)，进行仿真、分析、调试你的设计。该窗口可以显示设计的模块和模块，显示和设置断点、强制信号等。创建用户自己的按钮和执行经常使用的操作。

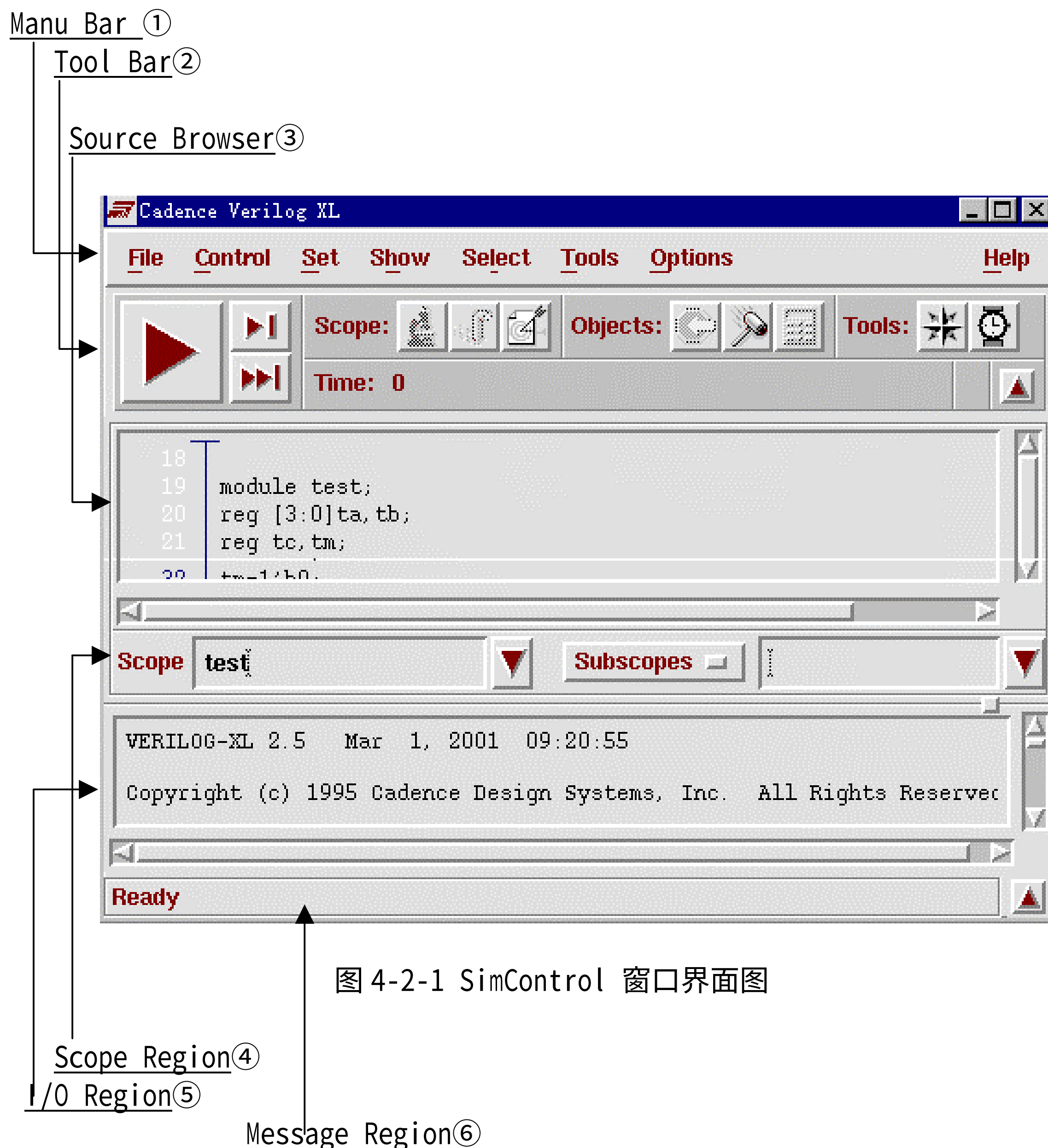


图 4-2-1 SimControl 窗口界面图

各部分简介：

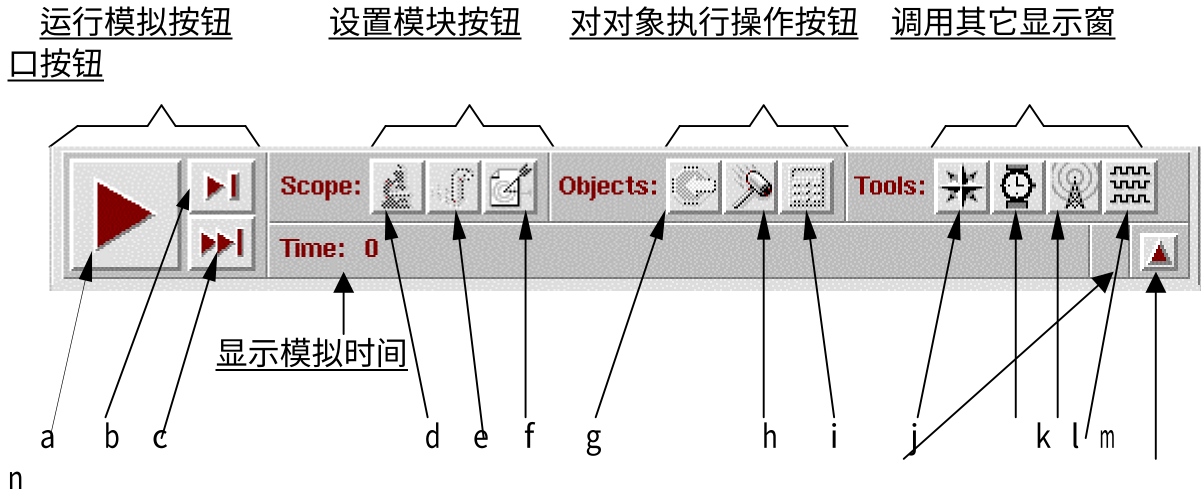
①、Menu Bar

有许多的子菜单，让你执行各种模拟仿真命令。这里就不一一介绍，到使用时，在指明其功能和所在位置。

②、Tool Bar

各种按钮代表最常用的操作和功能，能快速对选中的物体执行各种命令。你可以在工具条中加入自己定义的按钮，来代表常用的操作命令。使用

Option-User Buttons-Create 菜单项。用 Options-User Buttons-Edit 菜单项修改修改按钮。工具条还显示当前模拟时间，当处于交互式的模拟状态时，会随模拟更新时间。因为工具条按钮的操作为常用操作，下面各功能详细介绍一下。



放用户自定义按钮　是否显示程序代码

图 4-2-2 SimControl 窗口中的工具条

a、Run Simulation 按钮

运行模拟，若无断点直至完成，图标变为停止模拟图标。若有断点则运行到断点对应信号再改变的位置。

b、Single Step 按钮

再任何模块每按一下执行到下一个可执行行，即使在子程序中也是单步运行。

c、Step Over 按钮

在当前的模块中执行到下一个可执行行，在子程序中步单步执行，而是一步执行完子程序。

d、Set Scope 按钮

由当前的调试模块转到被选中的模块。

e、Scope Up 按钮

由当前模块转到它的上一级模块，但若有对象被选中，不执行。

f、Show Execution 按钮

模拟时更新当前模块，显示正在模拟的模块。在当前刚执行完的代码行左边有一个箭头

g、Set Breakpoint 按钮

设置断点，当模拟过程中被选信号变化时发生。代码左边的行号为高亮的可设为断点，灰色则不可以。

h、Set Force 按钮

弹出一个窗口，里面有当前选中信号的名字和数值。用户可以强制信号为一个希望值。

i、Show Value 按钮

n、程序代码是否显示的切换按钮。显示当前被选信号的数值。

以下 j、k、l、m 调用其它调试窗口，具体介绍放到后面。
j、打开 Navigator 窗口。
k、打开 Watch Objects 窗口。显示被选中的对象
l、打开 Singal Flow Browser 窗口。把被选中的对象放到浏览器中
m、打开 SimWave 窗口。显示被选中对象的模拟波形。
③、Source Browser
显示被调试的程序代码，每行左边有行号。你可以在其间选择信号和模块。这种选择会影响其它工具的操作对象,反过来其他工具操作对象的选择也会作用于 Source Browser 信号和对象的选择。可在其间设置断点，如前所说的在行号为高亮的行可设为断点，灰色则不可以。可在 Source Browser 中点鼠标的右键选择菜单进行操作。另一个对选择对象的操作是双击该对象。如双击信号得到它的数值,双击模块则调到该模块描述处。如图 4-2-2 中的 n 字母代表的按钮,Source Browser 可被关掉不显示。

④、Scope Region
包含 scope field 和 subscopes field。从下拉按钮选择不同的项，跳到不同的模块。对应的 Source Browser 显示该模块的代码。
⑤、I/O Region
显示执行的命令和模拟输出的结果。你也可以直接在此键入命令执行操作。I/O Region 也可以被关掉不显示，当点击 Message Region 右边的三角按钮可切换显示与否。
⑥、Message Region
显示模拟状态。

三．Navigator 窗口
按下和图 4-2-2 中 j 字母所代表的按钮一样的按钮打开 Navigator 窗口。此窗口用图形，在 Scope Tree 中采用树的形式显示设计中各模块的层次关系。在 Objects List 中显出 Scope Tree 中被选模块的当前模拟数值和描述。

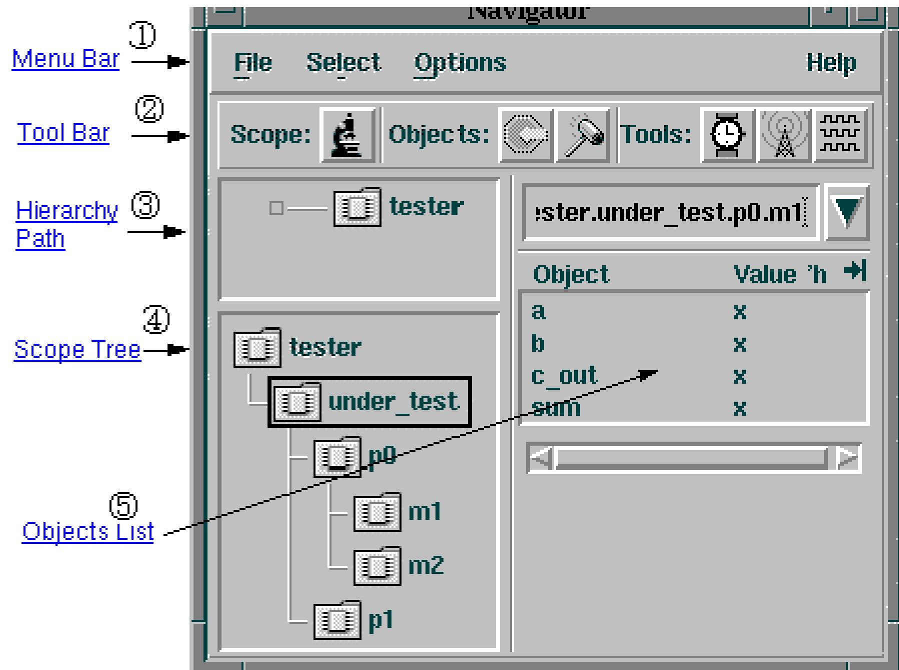


图 4-2-3 Navigator 窗口

①、Menu Bar

提供各种命令和操作，有下拉菜单(如下面的图 4-2-4)和右键弹出菜单两种。选中对象点击右键可选择对对象操作所需的命令，如下面的图 4-2-5。

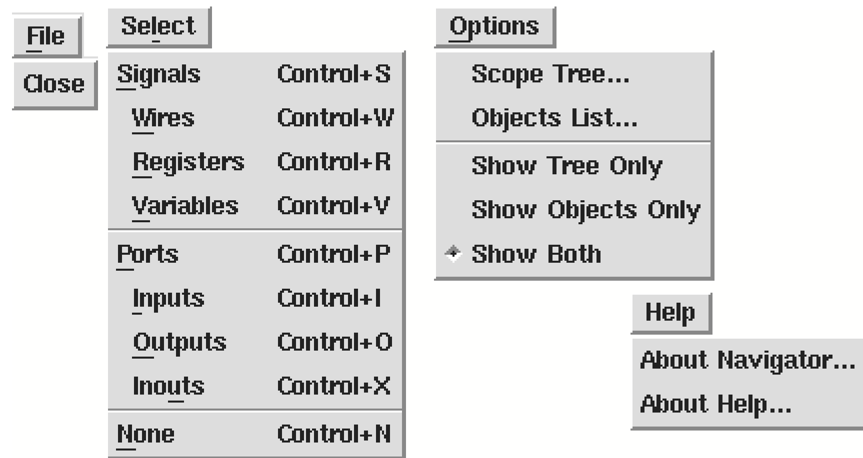


图 4-2-4 Navigator 窗口的菜单

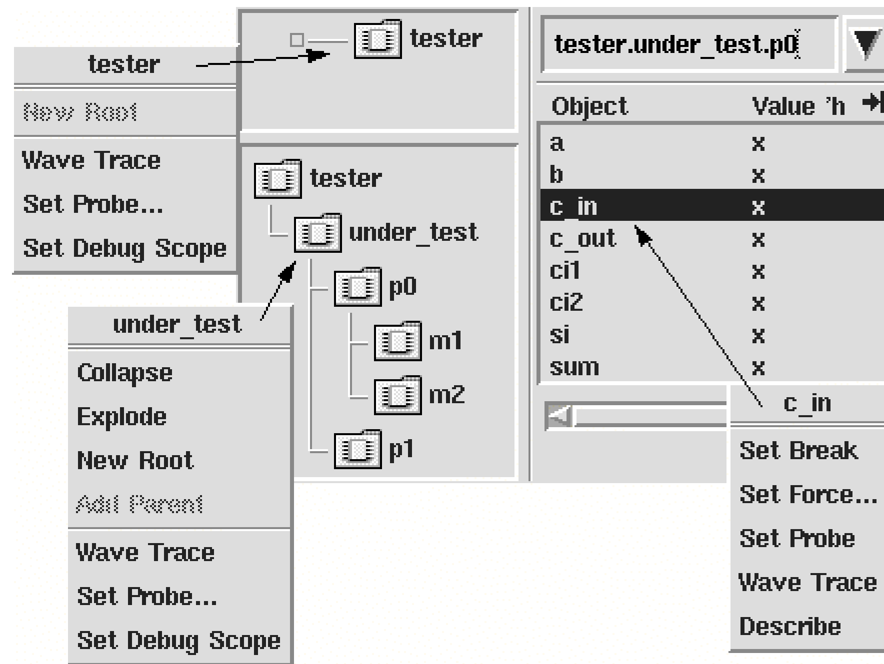
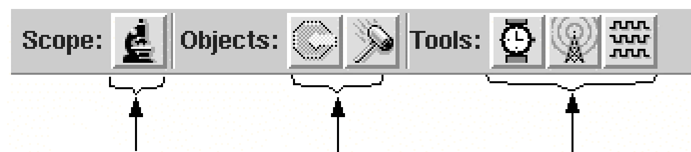


图 4-2-5 Navigator 中的 PoP-Up 菜单

②、Tool Bar



a、设置模块　b、对选择对象操作　c、调用其他显示窗口

图 4-2-6 Navigator 中的工具条

a、b、c 同 SimControl 窗口中的工具条对应按钮的功能一样，都是对选择对象进行相应的操作。只是对象可以在 SimControl 窗口选择也可以在 Navigator 窗口中选择，互相影响。

③ 、Hierarchy Path

显示当前模块的直接路径，其他路径不显示。可选择其间的模块点击右键弹出菜单进行操作。

④、Scope Tree

对被选中的模块用树的形式表示出来。在图 4-2-4 中 Options-Scope Tree...菜单项中有关于对象显示的的性质，有 Filters、Formatting、Layout 三栏，各有一些选项供选择。影响当前 Scope Tree 显示的内容。

⑤、Objects　List

显示当前调试模块里的信号和当前数值。在在图 4-2-4 中 Options-Objects List...菜单选项有 Filters、Formatting 两栏，会影响 Objects　List 中的显示内容。在 Selcet 子菜单中的选项(如图 4-2-4)能选取某一类别的信号，如都是 Wires 型，或是 Registers 型。

四. Singal Flow Browser 窗口

该窗口跟踪可疑信号的值，进入有三个方法

(1) 按下图 4-2-2 中 j 字母所代表的按钮

(2)SimControl 窗口 Tools- Singal Flow Browser 菜单项

(3) 图 4-2-6 Navigator 中的工具条中字母 c 的第二个按钮打开窗口。

(4) Wactch Objects Window 中按下图 4-2-2 中 j 字母所代表的按钮的一样的按钮界面如下图。(没选信号时)

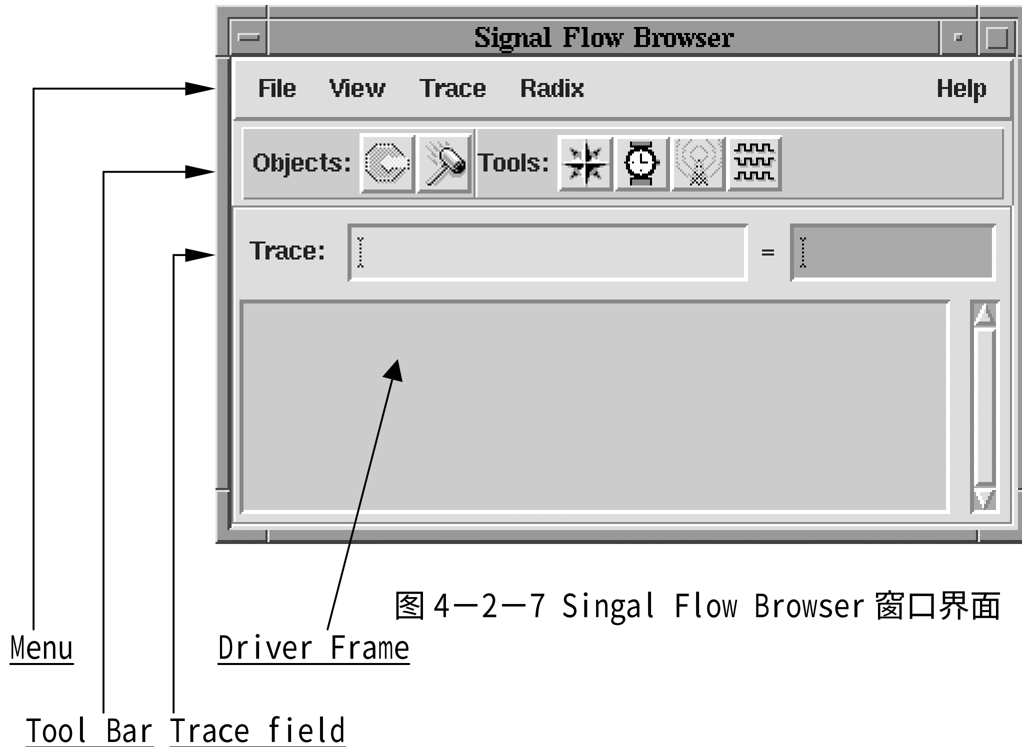


图 4—2—7 Singal Flow Browser 窗口界面

①、Menu

对对象的操作命令。可查看信号或输入的细节，显示信号的驱动，可用四种进制显示信号的数值见下图。后面会阐述菜单项的功能。

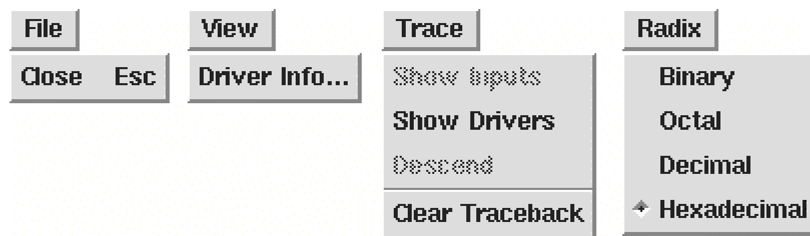


图 4—2—8 Singal Flow Browser 窗口菜单

②、Tool Bar 中的按钮和前面出现的相同的按钮的功能一样这里就不重复了。

③、Trace field 显示图 2 SimControl 窗口 Source Browser 或者图 4、Navigator 窗口中 Objects List 所选的信号。也可在 Trace field 输入信号名。

④、Driver Frame

显示被选的信号和数值，以及所有影响该信号的信号及它们的数值。
假设某个时候的 Driver 和 Value 如下图。

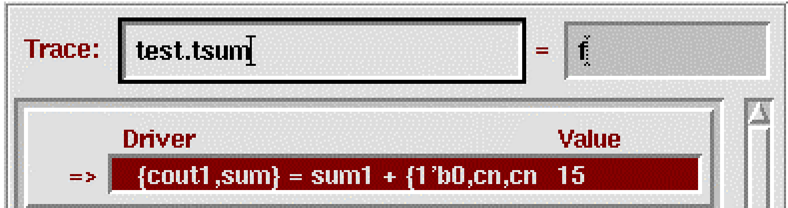


图 4-2-9 Driver 信号举例

如果在上图中选中 Driver 信号选图 4-2-9 中 View-Driver info...的菜单项，将弹出
Driver Details 窗口显示信号的详细信息。如下图。

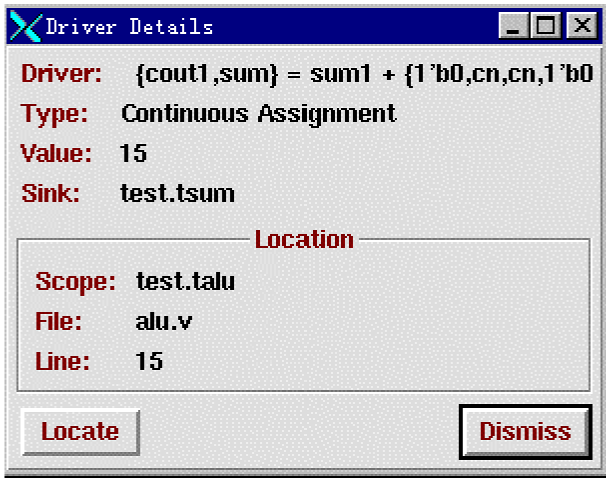


图 4—2—10 Driver 信号 Driver Details 窗口

当选中图 4-2-9 中的 Driver 信号，选图 4-2-8 中 Trace-show inputs 菜单项，或者双击信号，将得到影响 Driver 信号的有关信号的信息。如下图。

	Driver	Value
=>	{cout1,sum} = sum1 + {1'b0,cn,cn	15

Contributing Signal	Value
test.talu.sum1	f
Constant	0
test.talu.cn	1
test.talu.cn	1
Constant	0
Constant	00

图 4—2—11、Driver 信号的 inputs 信息图

再次双击 Driver 信号，会隐去这些信息。

五．Watch Objects 窗口

显示所选信号及其数值，当模拟中断时，更新数值。进入有三个方法

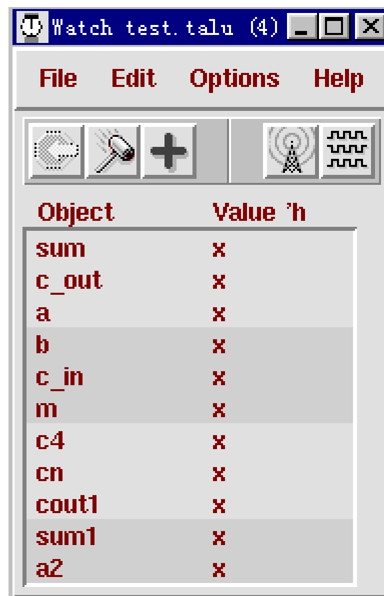
(1) 按下图 4-2-2 中 k 字母所代表的按钮

(2) SimControl 窗口 Tools- Watch Objects 菜单项

(3) Navigator 中的工具条中字母 c 的第一个按钮打开窗口。

(4) Singal Flow Browser

窗口中按下和图 4-4-2 中 k 字母所代表的按钮的一样的按钮界面如下图。(没选



信号时)

图 4—2—12、Watch Objects 窗口

你可以在打开 Watch Objects 窗口前选择观察信号，如在 Source Browser 中点选择信号，或在 SimControl 窗口中(图 2)的 Select 菜单下的菜单项选择，或在图 4 中 Navigator 窗口的⑤Objects List 中选择。也可以在打开 Watch Objects 窗口后再选择信号，如前选择好信号，然后点击图 4—2—12 中工具条上的加号图标，把选好信号加到窗口中。窗口的菜单如下图：菜单项的含义都比较明了，就不多说了。提一下 Options-Heighlight Activity 项使最新变化的信号项用高亮条表示，Options-Continous Update 项使信号随时变化，即使按图 3、中的 a、Run Simulation 按钮也会显示最后的结果，否则不显示最后结果。

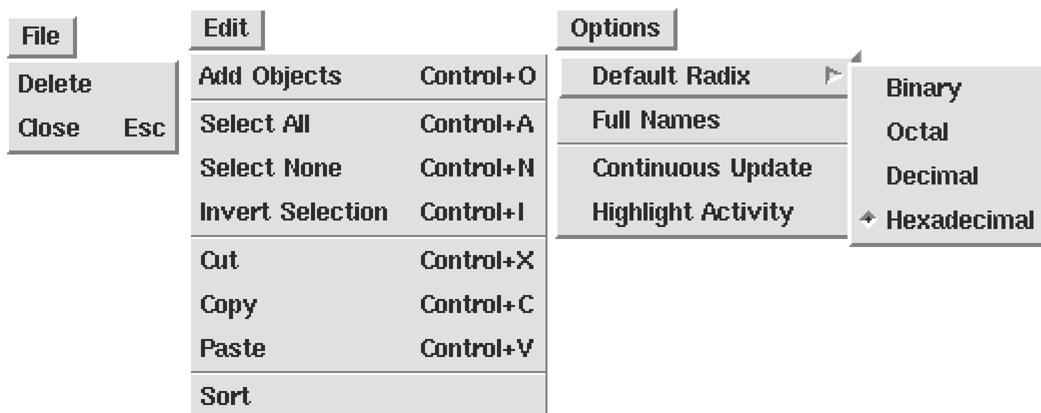


图 4—2—13、Watch Objects 窗口的菜单

⑥、SimWave 窗口
显示选择信号的波形和数值。

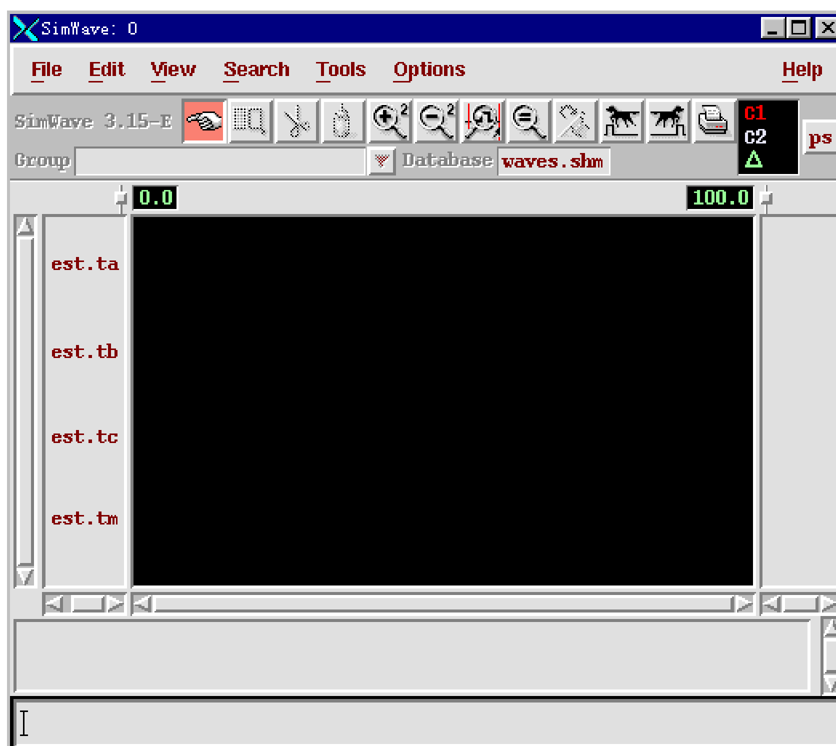


图 4—2—14、SimWave 窗口界面

§4—3 一个示例

这里举一个实际工作中编的例子，演示前面所讲的内容，但不一定面面俱到。程序的清单见附录。(alu.v)

①、在命令行中敲 textedit alu.v↙ 用 textedit 编好程序的文本。

②、在命令行中敲 verilog –c alu.v↙ 编译通过程序.

③、在命令行中敲 verilog –s alu.v +gui&↙ 进入交互式图形界面 SimControl 窗口。(见图 2)在 Scope 中选择 test.talu

④、在 SimControl 窗口中的选中 Select-Ports 项，选择端口。

⑤、按下图 3、SimControl 窗口中的工具条中的 k 键，打开 Watch Objects 窗口，
并如图 13 选中 Options-Continuous ,Highlight Activity 两项。

⑥、按下图 3、SimControl 窗口中的工具条中的 m 键,打开 SimWave 窗口。

⑦、按下图 3、SimControl 窗口中的工具条中的 a 键,

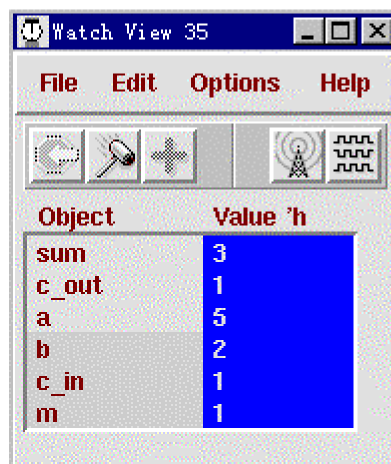


图 4—3—1、Watch Objects 窗口

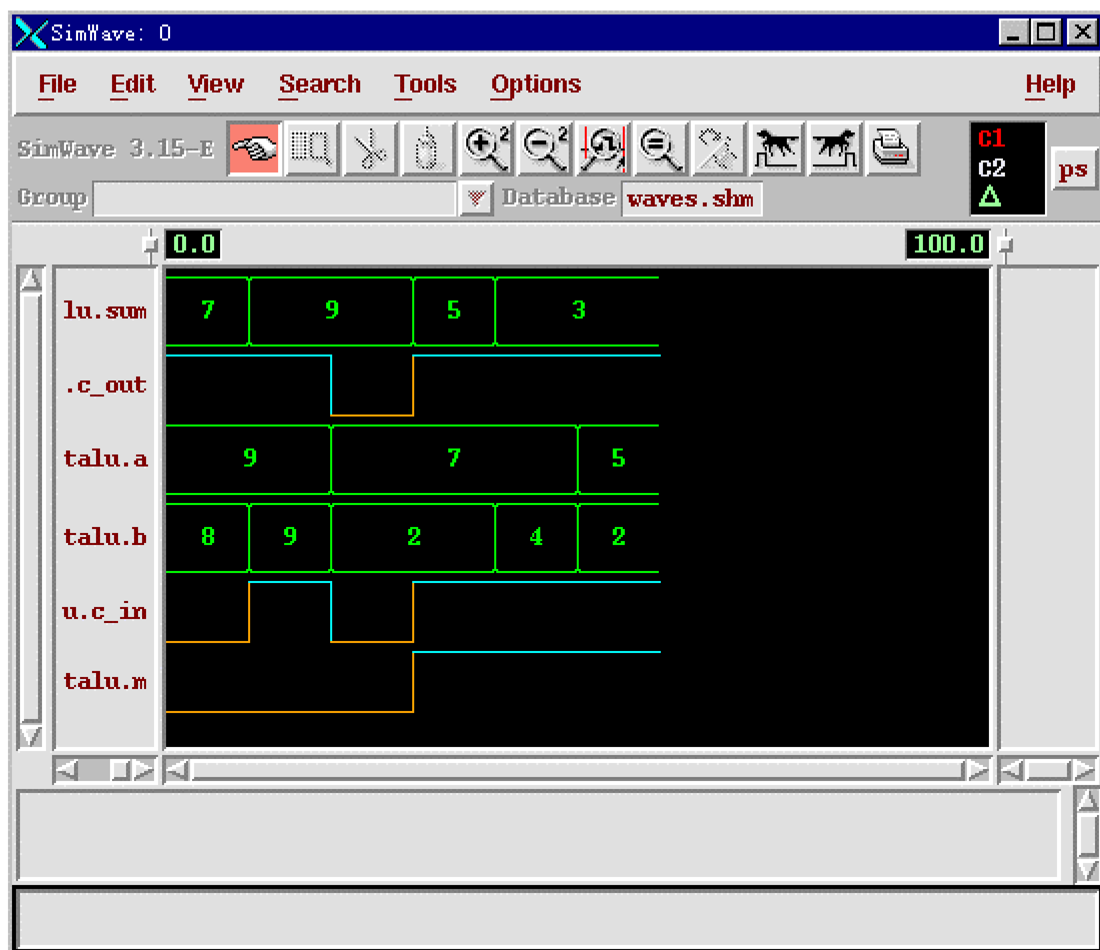


图 4—3—2、SimWave 窗口波形

附：alu.v 源程序：

```
module alu(sum,c_out,a,b,c_in,m);
output [3:0]sum;
output c_out;
input [3:0]a,b;
```

```
input c_in,m;
wire c4,cn,cout1;
wire [3:0]sum1,a2;
assign a2[0]=(b[0]&~m)|(~b[0]&m);
assign a2[1]=b[1];
assign a2[2]=(b[2]&~m)|(((~b[2]&b[1])|(b[2]&~b[1]))&m);
assign a2[3]=(b[3]&~m)|(~b[3]&~b[2]&~b[1]&m);
assign {c4,sum1}=a+a2+c_in;
assign cn=c4|(sum1[3]&sum1[2])|(sum1[3]&sum1[1]);
assign {cout1,sum}=sum1+{1'b0,cn,cn,1'b0}+1'b0;
assign c_out=cn;
endmodule
module test;
reg [3:0]ta,tb;
reg tc,tm;
wire [3:0]tsum;
wire tcout;
alu  talu(tsum,tcout,ta,tb,tc,tm);
initial
$monitor($time,"c_out=%d,sum
%d=%d+%d+%d,m=%d",tcout,tsum,ta,tb,tc,tm);
initial
begin
ta=4'b1001;
tb=4'b1000;
tc=1'b0;
tm=1'b0;
#10 ta=4'b1001;
tb=4'b1001;
tc=1'b1;
#10 ta=4'b0111;
tb=4'b0010;
tc=1'b0;
#10 tm=1'b1;
ta=4'b0111;
tb=4'b0010;
tc=1'b1;
#10 ta=4'b0111;
tb=4'b0100;
tc=1'b1;
#10 ta=4'b0101;
tb=4'b0010;
tc=1'b1;
#10 $finish;
```

```
end
endmodule
```

下图 17、18 是程序对应的电路图。

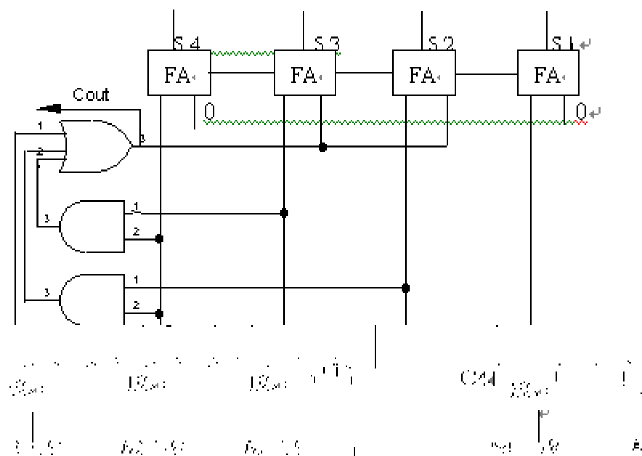


图 17　BCD 码加法器

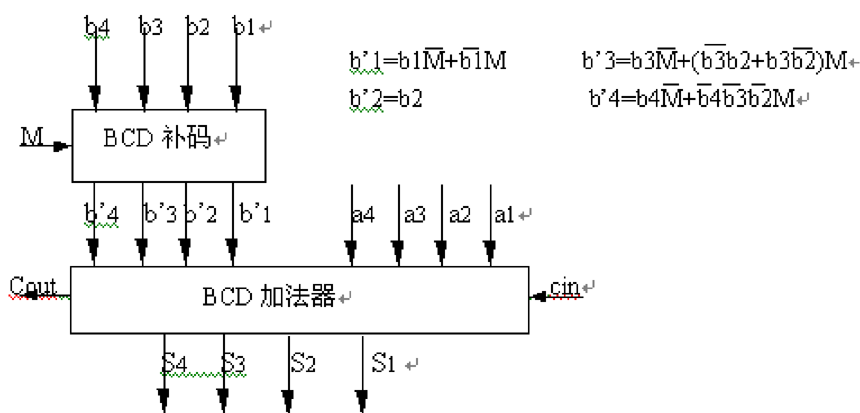


图 18　ALU 原理图

几个打开相关帮助的命令,在命令行中敲入：

openbook vlogtut&↙　　(Verilog-XL Tutorial)
openbook vlogref&↙　　(Verilog-XL Reference)
openbook vloguser&↙　　(simwave user guide)
openbook simwaveuser&↙ (open the Verilog-XL guide)

校内网站(ftp :10.12.41.35)有 PC 机版 Verilog 仿真工具如 Modelshim,Active HDL4.2 版，Xilinx 的 FPGA 等等，还有 Cadence 的一些资料（如 Verilog-XL Reference、user_guide 等等)。

第五章 MEDICI 的使用简介

§5－1 MEDICI 的使用

1 登陆

请使用自己的帐号登陆到 10.13.72.33。

进入系统后,请进入目录/export/home/avant/bin

如果用户是从 PC 机上登录的，应确保在登录前已经运行了 EXCEED（该程序可以使工作站上的图形输出在 PC 机上可见），在登陆后应接着运行命令（这个命令使得程序结果输出在指定的 PC 机上）：

setenv DISPLAY 本机 IP：0.0

2 运行 MEDICI

步骤如下：将所有的语句写在一个文本文件中，在运行命令 medici 后系统将提示是否输入文件名，此时可以将文件名输入。或者直接将文件名作为参数和 medici 命令一起运行如下：

medici /export/home/avant/public/study.txt <回车>

§5－2 关于 MEDICI 的概述

关于 MEDICI 语法的详细描述请参阅使用手册（Manule.pdf），在该手册中有几种不同类型结构的例子（如 MOS 和 NPN），请结合例子来准确理解语句的用途。

一．MEDICI 的功能简介

Medici 是先驱（AVANT！）公司的一个用来进行二维器件模拟的软件，它对势能场和载流子的二维分布建模，通过解泊松方程和电子、空穴的电流连续性等方程来获取特定偏置下的电学特性。用该软件可以对双极型、MOS 型等半导体器件进行模拟，这个程序通过解二极管和双极型三极管以及和双载流子有关的电流效应（诸如闩锁效应）的电流连续性方程和泊松方程来分析器件。Medici 也能分析单载流子起主要作用的器件，例如：MOSFET，JFET，MESFET。另外，MEDICI 还可以被用来分析器件在瞬态情况下的变化。在亚微米器件模拟中，MEDICI 通过联解电子和空穴的能量平衡和其他的器件方程，可以对深亚微米的器件进行模拟。像热载流子和速度过冲等效应在 MEDICI 中都已经考虑了，并能够对他们的影响进行分析。

二．MEDICI 的一些特性

网格（GRID）

在 MEDICI 使用了非均匀的三角形网格，可以处理具有平面和非平面表面的特殊器件，并且能够根据电势或杂质分布的情况自动进行优化。电极可以被放在器件结构中的任何地方。

杂质分布的读入

杂质的分布可以通过 MEDICI 的函数从 AVANT！的其他工艺建模软件如：TMA SUPREM3 和 SUPREM4 或者是包含杂质分布的文本文件中获得，也可以在文本文件中描述。

物理模型

为了使模拟的结果精确，下列模型都可以被考虑进来：载流子的复合，PHOTOGENERRATION，碰撞离化效应，禁带变窄效应，BAND-BAND TUNNELING，迁移率的变化，载流子寿命，载流子的 Boltzman 和 Fermi-Dirac 统计分布，部分离化效应。

其他特性

1．Attach lumped resistive，capacitive，and inductive elements to contacts
2．可以描述分布式接触电阻
3．可以在模拟中描述电压和电流的边界条件
4．I-V 曲线自动跟踪
5．为了计算和频率相关的电容，电导，admittance 和 s 参数，可以在任何虚拟的频率下进行交流小信号分析

图形的输出

1．One-dimensional plots of terminal data
可以用来显示直流特性，例如，所加的电压，接触端的电压，终端电流，时间（瞬态特性），还能够用来显示交流量，如电容，电导，admittance，频率，以及用户定义的一些变量。
2．可以显示沿着器件结构中特定路径上的某一参量的一维分布
包括：势能，载流子的准费米势能，电场，载流子浓度，杂质浓度，复合和产生率，以及电流密度。
3．网格，边界，电极，和结的位置，耗尽区边界的二维结构图
4．量的二维图形分布，例如：势能，载流子的准费米势能，电场，载流子浓度，杂质浓度，复合和产生率，电流密度，电流分布。
5．电流密度核电场的二维向量分布
6．Three-dimensional projuction plots of quantities，例如：势能，载流子的准费米势能，电场，载流子浓度，杂质浓度，复合和产生率，电流密度，

三．MEDICI 的语法概览

语句简介

器件结构定义语句包括:

MESH, X.MESH, Y.MESH, ELIMINATE, SPREAD, BOUNDRY, TSUPREM4,REGION， ELECTRODE, PROFILE。这些语句定义了器件的结构和模拟用的网格.

MESH:初始化网表的生成
X.MESH:描述 X 方向上的网格线的位置
Y.MESH:描述 Y 方向上的网格线的位置
ELIMINATE:沿着网格线缩减节点
SPREAD:沿着水平网格线调整节点的垂直位置
BOUNDRY:调整模拟的网表以适应边界的界面

REGRID 可以用来用来对这种网格进一步优化.

材料物理性能描述

REGION:描述材料在结构中的区域
INTERFACE 语句可以被用来说明界面层电荷,陷阱,和复合速率.
CONTACT 被用来说明电极边上的特殊边界条件.
MATERIAL 可以用来改变结构的材料特性.

器件求解的物理模型

MOBILITY 描述和各种各样的迁移率模型相关的参数.
MODELS 用来描述模拟过程中的物理模型.
SYBOLIC 可用来选择模拟时用的求解方法.
METHOD 用来对特定的求解方法选择特殊的技巧.
SOLVE 用来选择偏置条件和分析类型.,这个语句可以被用于稳态,瞬态和交流小信号.

图形化结果的输出

PLOT.3D 被用来初始化三维图显示平台，它的配套语 3D.SURFACE,TITLECOMMENT 等.

PLOT.2D 用来初始化二维图形显示平台.它的配套语句可以有 CONTOUR,VERCTOR,E.LINE,LABEL,TITLE,COMMENT 等.

PLOT.1D 用来初始化一维图形显示平台,它的配套语句有 E.LINE,LABEL,TITIE,COMMEN，CONTOUR 等.

网表描述的步骤

通常,网表的描述有以下步骤:

1. 定义一系列有间隔的 X 和 Y 方向的网格线构成的一个简单的矩形
2. 将网格线适当扭曲以适应非平面的图形或者与杂质的分布相匹配(平面性很差的结构很难处理好)，这一步的目的是为了将网格进行优化。
3. 将多余的节点从网格中去除掉。
4. 描述材料区域和电极

语句格式:

MEDICI 的输入语句具有自由的格式,并具有下列的一些特性.

a) 每一个语句都由语句名称开始,后面再跟一些参数名和值.
b) 每一个语句都可以占用一行以上的地方,行与行之间用连接符号(“+”)连接.
c) 每一行最多由 80 个字符构成

参数类型:

参数是指接在每一个语句名称后，用来定量的实现该语句的功能的符号。

a) logical:如果该参数出现,则表示为 true
b) numerical
c) array
d) character

输入限制:

1. 最多 1000 个语句
2. 最多 2000 行
3. 最多 60000 个字符

§5－3　教学实例 1

这里以一个 NMOS 为例作了一些分析.关于这个例子的描述文件放在 /export/home/avant/public/study.txt 中，可以通过 FTP 将这个文件下载（这是一个文本文件），假如想在这个文件的基础上做进一步的修改的话，修改后必须将文件上载到原来的目录下，然后按照前面所说的方法运行该文件。另外使用手册也放在这个目录下，有兴趣的话可以自己下载了去看（用 Acrobat 打开）。

请各位需要注意的是，在使用之前请每人自己作一个备份，并使用另外的名字，以免因为使用同一个文件名致使程序运行到半中间时产生冲突而中断。

例子如下;

```
TITLE     TMA MEDICI Example 1 - 1.5 Micron N-Channel MOSFET
```

给本例子取的标题，对实际的模拟无用

```
COMMENT   Specify a rectangular mesh
```

COMMENT语句表示该行是注释

```
MESH      SMOOTH=1
```

创建器件结构的第一步是定义一个初始的网表(见图1),在这一步中网表不需要定义得足够精确,只需要能够说明器件的不同区域,在后面我们会对该网表进行优化.网表的生成是由一个MESH语句开始的,MESH语句中还可以对smoothing进行设置(好的smoothing可以把SPREAD语句产生的钝角三角形带来的不利影响减小).

```
COMMENT   WIDTH is the whole width,H1 is the width of a grid
X.MESH    WIDTH=3.0  H1=0.125
```

X.MESH和Y.MESH语句描述了初始网表是怎样生成的,X.MESH用来描述横向的区域.在此例子中,X.MESH语句中的H1=0.125说明在横向区域0—WIDTH之间网格线水平间隔为0.125微米(均匀分布).

```
COMMENT   location of line NO. 1 is -0.025u, No.3 is 0.0u
Y.MESH    N=1   L=-0.025
```

Y.MESH用来描述纵向的区域,在这参数N指第一条水平网格线,L指位于－0.025微米处

```
Y.MESH    N=3   L=0.
```

第三条水平线位于0微米处

在这个例子中头三条水平线用来定义厚度为0.025微米的二氧化硅(栅氧).

```
COMMENT   0u-1.0u H1=0.125  1u-2u H1=0.250
Y.MESH    DEPTH=1.0  H1=0.125
```

这条语句添加了一个1微米深(DEPTH)的,垂直向网格线均匀间隔0.125微米(H1)的区域

```
Y.MESH    DEPTH=1.0  H1=0.250
```

添加了一个 1 微米深的,垂直向网格线均匀间隔 0.250 微米的区域

```
COMMENT    Eliminate some unnecessary substrate nodes
ELIMIN     COLUMNS   Y.MIN=1.1
```

该语句将 1.1 微米(Y.MIN)以下的网格线隔列(COLUMNS)删除,以减小节点数

```
COMMENT    distort source/drain oxide thickness using SPREAD
COMMENT    ENC means the abruptness of two reagion, the number more litter,more sharp
SPREAD     LEFT   WIDTH=.625   UP=1   LO=3   THICK=.1   ENC=2
```

SPREAD语句用来对网格线进行扭曲,以便更好的描述器件的边界.这个SPREAD语句将前三条网格线在左边(0-WIDTH之内)的间隔从0.025(栅区氧化层)过渡到0.1微米(源区氧化层).其中UP指要定义的区域的上边界(此处为第一条网格线),LO指要定义的区域的下边界(此处为第三条网格线),THICK定义了这个区域的厚度.

```
SPREAD     RIGHT  WIDTH=.625   UP=1   LO=3   THICK=.1   ENC=2
```

这个SPREAD语句将前三条网格线的在右边的间隔从0.025(栅区氧化层)过渡到0.1微米(漏区氧化层).

参数ENC决定了从厚的区域过渡到薄的区域的变化特性.值越大过渡区越平缓,可以自己修改这个参数,看看过渡区有什么变化（ENC=2表明只在两格完成过渡）

WIDTH在这里以过渡区域的中点为准。

```
COMMENT    Use SPREAD again to prevent substrate grid distortion
COMMENT    line NO.4 move to Y.Lo, >line No.4 will be not affected
SPREAD     LEFT   WIDTH=100    UP=3   LO=4   Y.LO=0.125
```

这个 SPREAD 语句将第四条网格线固定在 0.125 微米处（Y.LO=0.125）,可以使前两条 SPREAD 语句产生的网格扭曲不影响到 0.125 微米以下的网格

在这儿WIDTH参数取了一个特别大的值,可以把过渡性的区域放在器件的外面.

```
COMMENT    Specify oxide and silicon regions
COMMENT    no more description means all reagion
REGION     SILICON
```

REGION是用来定义区域的材料性质,如果不特别说明区域的范围的话,则表示对整个结构进行定义,在这里定义整个区域为硅

```
REGION     OXIDE     IY.MAX=3
```

定义第三条网格线以上的区域为二氧化硅

```
COMMENT    Electrode definition
ELECTR     NAME=Gate     X.MIN=0.625   X.MAX=2.375   TOP
```

ELECTR是用来定义电极位置的,在这里将栅极放在栅极二氧化硅的表面

```
ELECTR     NAME=Substrate   BOTTOM
```

将衬底接触电极放在器件的底部

```
ELECTR     NAME=Source   X.MAX=0.5   IY.MAX=3
```

将源区的接触电极放在器件的左边

```
ELECTR     NAME=Drain    X.MIN=2.5   IY.MAX=3
```

将漏区的接触电极放在器件的右边

```
COMMENT   Specify impurity profiles and fixed charge
PROFILE   P-TYPE  N.PEAK=3E15  UNIFORM
```

PROFILE语句是用来定义掺杂情况的,P-TYPE表示是P型掺杂,N.PEAK描述峰值浓度.

这个语句定义整个衬底的浓度为均匀掺杂(UNIFORM),浓度为P型(P-TYPE)3E15(N.PEAK).

```
PROFILE   P-TYPE  N.PEAK=2E16  Y.CHAR=.25
```

这个语句定义沟道阈值调整的掺杂为P型,浓度为2E16,掺杂的特征长度(Y.CHAR)为0.25微米

```
PROFILE   N-TYPE  N.PEAK=2E20  Y.JUNC=.34  X.MIN=0.0  WIDTH=.5   XY.RAT=.75
PROFILE   N-TYPE  N.PEAK=2E20  Y.JUNC=.34  X.MIN=2.5  WIDTH=.5   XY.RAT=.75
```

以上两句定义了源(0-0.5微米处)和漏(2.5-3微米处)的掺杂区,他们的结深(Y.JUNC)为0.34微米,横向扩散率为0.75(XY.RAT),为N型(N-TYPE),浓度为2E20(N.PEAK).

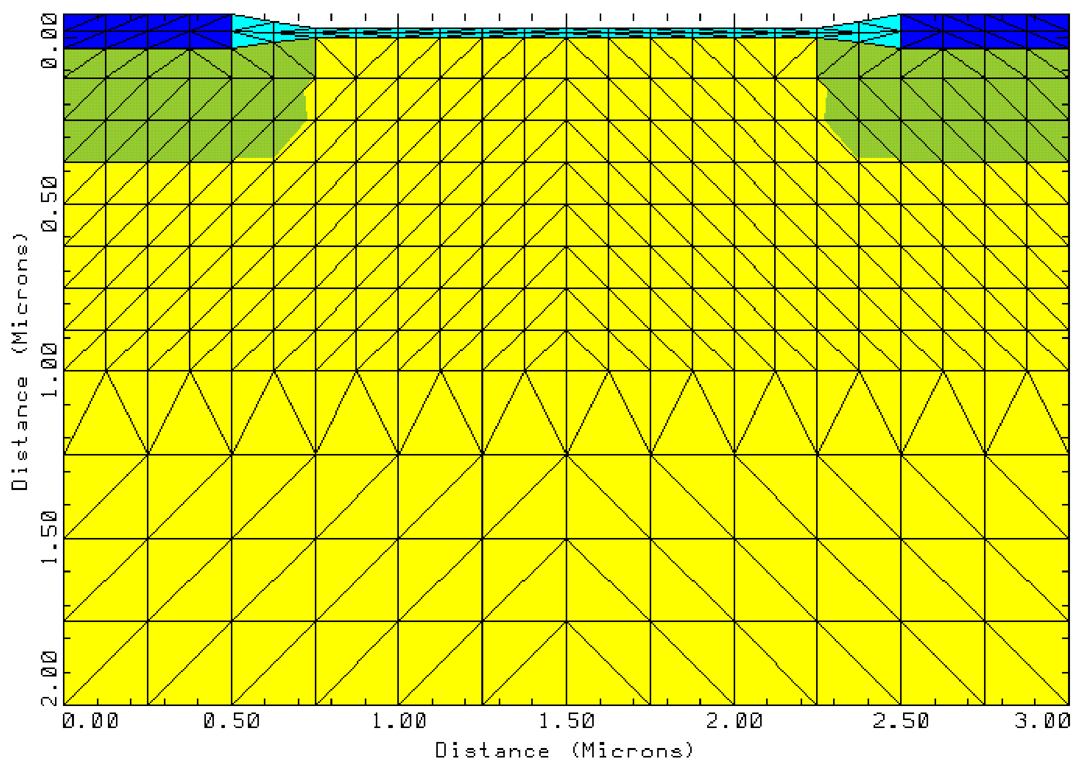
```
INTERFAC  QF=1E10
```

INTERFAC语句是用来定义界面态的,这个语句说明在整个二氧化硅的表面有浓度一致的固定态,浓度为1E10(QF).

```
COMMENT   GRID means show/hide grid
+         FILL means reagions is color filled or not
+         SCALE means the plot is reduced from the specified size in x or y directions
PLOT.2D   GRID  TITLE="Example 1 - Initial Grid"   FILL  SCALE
```

PLOT.2D是用来显示二维图形的语句,参数GRID表示在图中显示网表,FILL表示不同的区域用颜色填充，使用参数SCALE后,可以使显示图形的大小合适.这个语句本身并不能显示器件的什么特性,只是给器件特性的显示提供一个平台,结合了其他的语句后才能显示所想要的图形,这一点在下面会给出示范.在这里的几个参数都是可有可无的,不妨把他们去掉,看看有什么不同,以加深理解.

该语句所得的图形如下:



到目前为止,器件的结构已经定义了,下面将对该网格进行调整以适应模拟的需要.

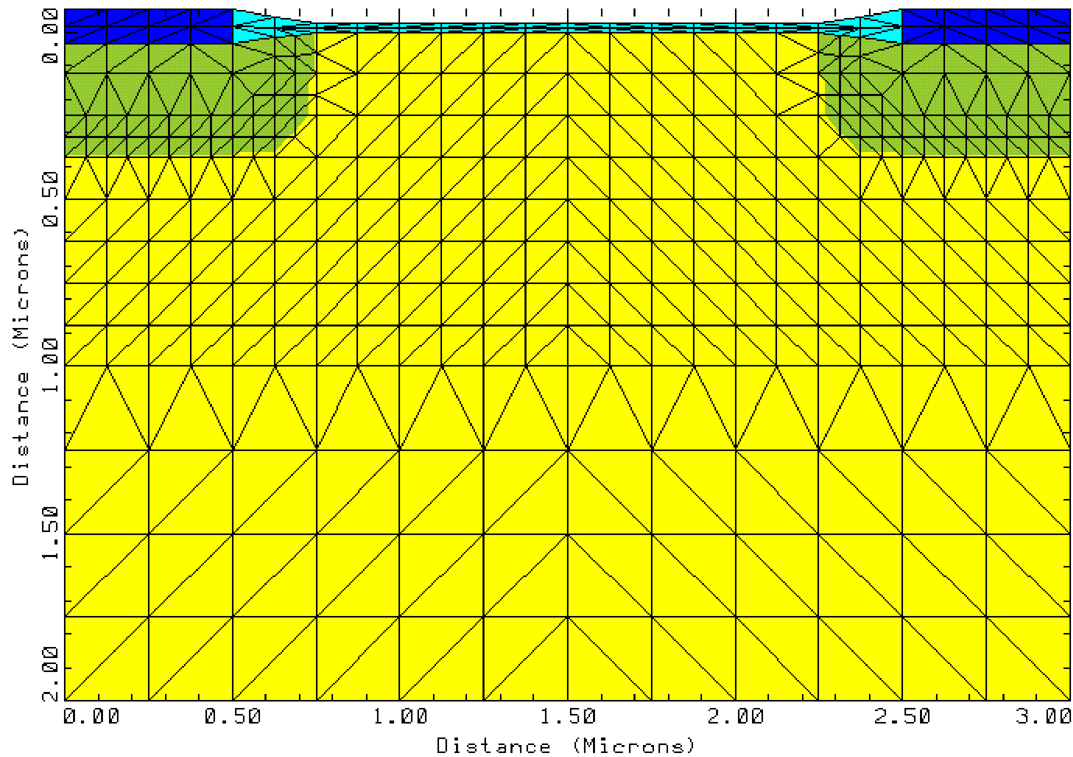
```
COMMENT    Regrid on doping
REGRID     DOPING    IGNORE=OXIDE   RATIO=2   SMOOTH=1
```

REGRID语句是用来对网格按要求进行优化的语句.

当节点的掺杂特性超出了RATIO的要求时,该三角形网格将被分割成四个适合的小三角形,但二氧化硅区域不被包含在内(由IGNORE说明).SMOOTH用来平滑网格的,以减小钝角三角形带来的不利影响,SMOOTH=1表示平滑湖化时,各个区域的边界不变,SMOOTH=2表示仅仅不同材料的边界保持不变.参数DOPING说明优化网格的标准是基于杂质分布的,杂质分布变化快的区域自动进行调整.

```
PLOT.2D    GRID  TITLE="Example 1 - Doping Regrid"   FILL   SCALE
```

该语句生成的图形如下,大家可以仔细比较一下和上图的区别(在网格上有什么不同,尤其是在PN结的边缘.这儿浓度的变化最快).



COMMENT   Specify contact parameters
CONTACT   NAME=Gate   N.POLY

CONTACT语句是用来定义电极相关的一些物理参数,在这儿栅极(NAME)的材料被定义为N型的多晶硅(N.POLY).

COMMENT   Specify physical models to use
MODELS   CONMOB   FLDMOB   SRFMOB2

MODELS用来描述在模拟中用到的各种物理模型,模拟时的温度也可以在这里设定(由参数TEMP设定).除非又使用了该语句,否则该语句定义的模型一直有效.

参数CONMOB表示使用迁移率与杂质分布有关的模型，参数FLDMOB表示使用迁移率与电场分布有关的模型.参数SRFMOB2表示表面迁移率降低效应将被考虑.

COMMENT   Symbolic factorization, solve, regrid on potential
SYMB   CARRIERS=0

The SYMBOLIC statements performs a symbolic factorization in preparation for the LU decompositions in the solution phase of the program.

在这儿只选用了Poisson来解方程,因为在这只需要势能,所以载流子类型为零.

COMMENT
METHOD   ICCG   DAMPED

METHOD语句设置了一个和SYMB语句相关的特定的求解的算法

在大多数的情况下,只需要这两个参数就能够得到最有效的零类型载流子模拟.

SOLVE

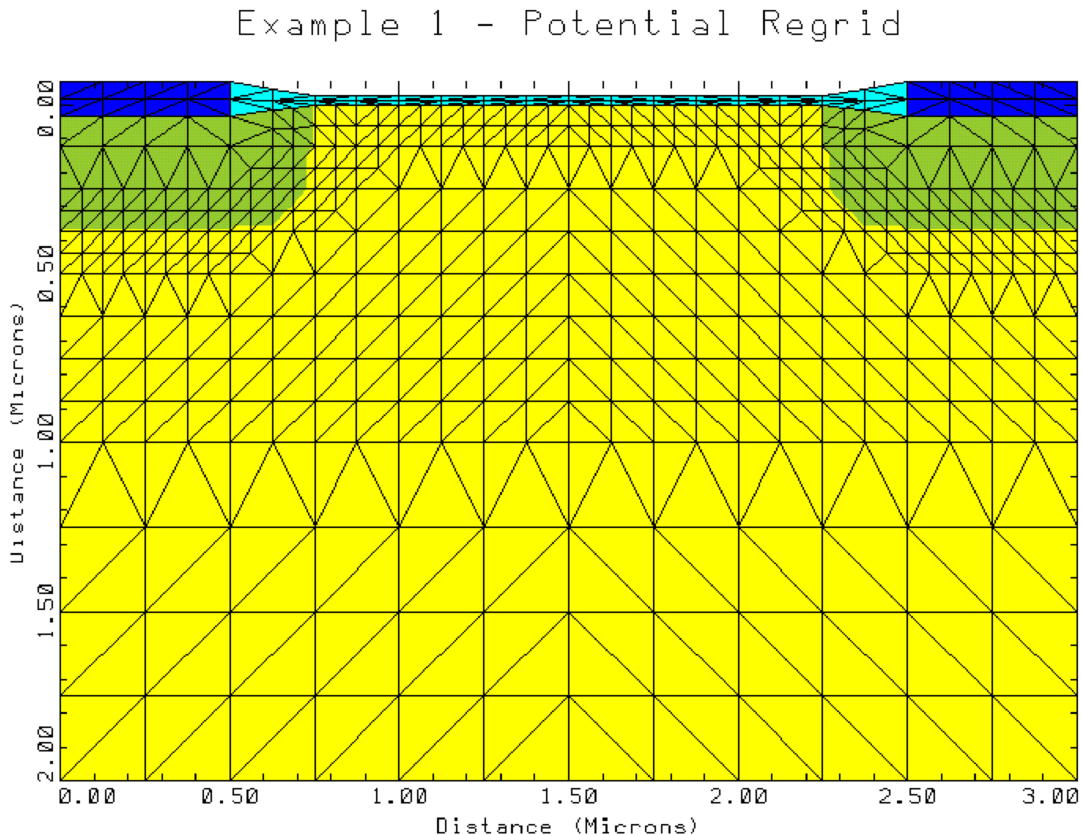
该语句用来获得解,在这里初始条件设置为0

REGRID POTEN IGNORE=OXIDE RATIO=.2 MAX=1 SMOOTH=1

该语句可以在势能变化快的地方将网格进一步优化,

PLOT.2D GRID TITLE="Example 1 - Potential Regrid" FILL SCALE

该语句显示的图形如下：



COMMENT Impurity profile plots

PLOT.1D DOPING X.START=.25 X.END=.25 Y.START=0 Y.END=2

\+ Y.LOG POINTS BOT=1E15 TOP=1E21 COLOR=2

\+ TITLE="Example 1 - Source Impurity Profile"

PLOT.1D语句是用来显示参数的一维变化的.在这里参数DOPING说明显示的是杂质的分布情况,X.START,X.END,Y.START,Y.END用来定义想要考察的路径（起始坐标是（X.START,Y.START）,终点坐标是（X.END,Y.END））.

Y.LOG 表示纵坐标使用对数坐标,最大值为 TOP,最小值为 BOT.

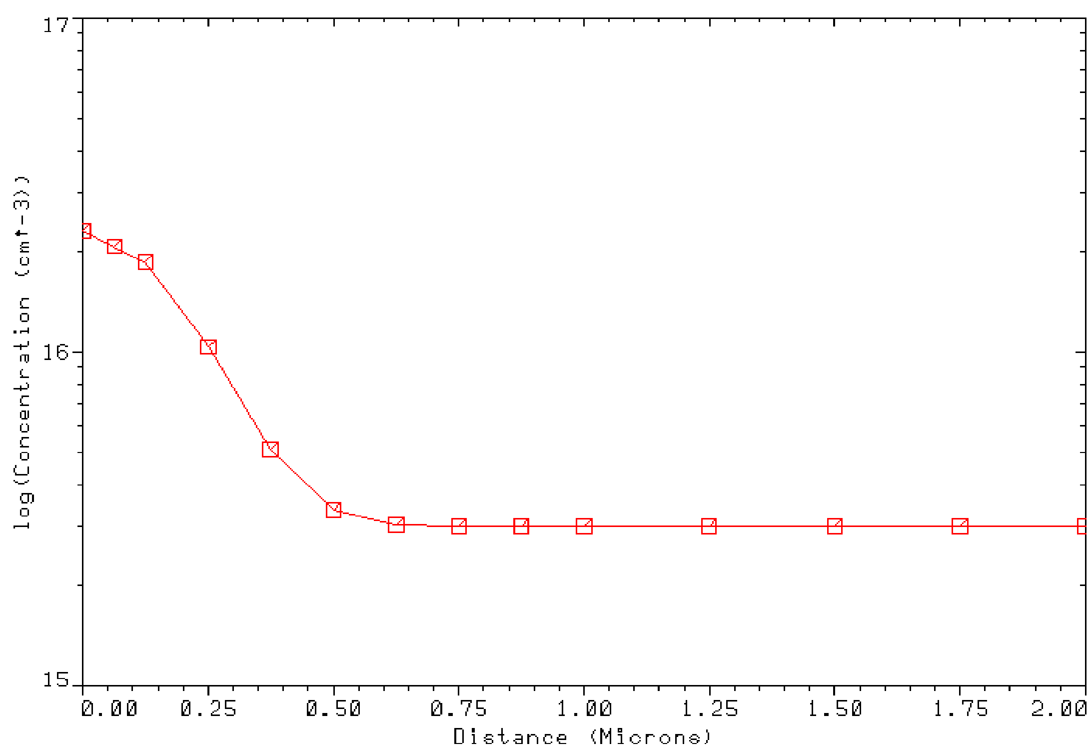
参数 COLOR 用来描述该曲线选用的颜色，不妨改变该参数，看看颜色发生了什么变化。

这条语句用来显示从（0.25,0）到（0.25,2）上的一维杂质分布,具体结果见图：




```
PLOT.1D    DOPING  X.START=1.5  X.END=1.5  Y.START=0  Y.END=2
+          Y.LOG  POINTS  BOT=1E15  TOP=1E17  COLOR=2
+          TITLE="Example 1 - Gate Impurity Profile"
```

这条语句用来显示从（1.5,0）到（1.5,2）上的一维杂质分布，具体结果见图：



```
PLOT.2D   BOUND  REGION TITLE="Example 1 - Impurity Contours"  FILL  SCALE
CONTOUR   DOPING  LOG  MIN=16   MAX=20   DEL=.5  COLOR=2
CONTOUR   DOPING  LOG  MIN=-16  MAX=-15  DEL=.5  COLOR=1  LINE=2
```

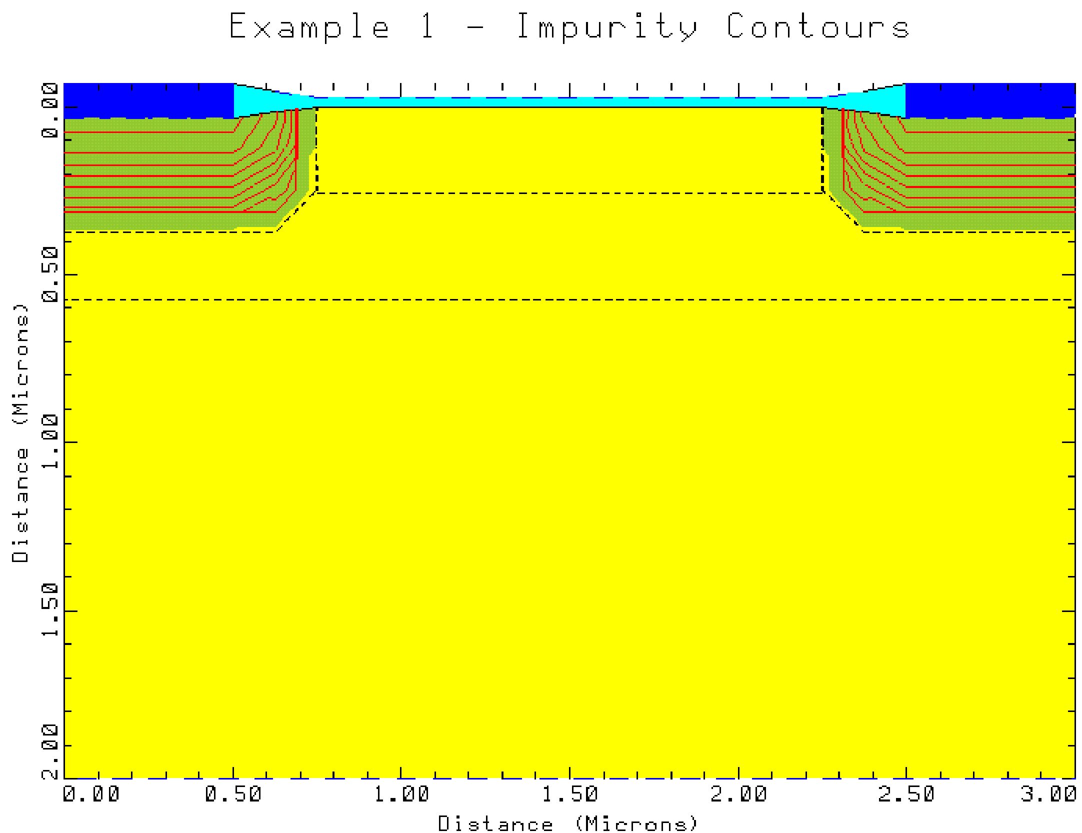
在这里PLOT.2D语句搭建了一个显示的平台,两个CONTOUR语句则在这个平台上描绘了所需参数的特性,CONTOUR语句是用来在最近的一个PLOT.2D语句上绘制各种物理参量的二维特性的，在这里它们都是用来绘制杂质的二维分布（由参数DOPING说明），不妨把其中一个语句去掉，看看该语句产生的曲线是那些。

MIN和MAX则指定了参数的显示范围，DEL表示所显示的相邻曲线之间的在参数值上的间隔，负数表示是P型掺杂，正数表示是N型掺杂。

COLOR表示线条的颜色，LINE表示线条的类型，不妨把这两个参数改变一下，看看对应的是那一个曲线。

LOG表示MIN，MAX和DEL都采用对数表示。

上面三条语句产生的图形如下：




```
COMMENT   Solve using the refined grid, save solution for later use
SYMB      CARRIERS=0
SOLVE
```

为了给下面的模拟提供一个起始条件，在这获得了一个零偏置解

```
COMMENT   Do a Poisson solve only to bias the gate
SYMB      CARRIERS=0
```

```
METHOD     ICCG  DAMPED
SOLVE      V(Gate)=1.0
```

在使用SOLVE语句获得下一个解之前，SYMB语句必须再使用一次。因为网表的节点数在上一次求解的时候已经改变。因为器件在零偏置的时候，电流很小，所以使用零载流子模型就足够了。

```
COMMENT    Use Newton's method and solve for electrons
SYMB       NEWTON  CARRIERS=1  ELECTRON
```

下面将要求解漏极电压和漏极电流的关系，因为是NMOS器件，所以设置载流子类型为电子

```
COMMENT    Ramp the drain
SOLVE      V(Drain)=0.0  ELEC=Drain  VSTEP=.2  NSTEP=15
```

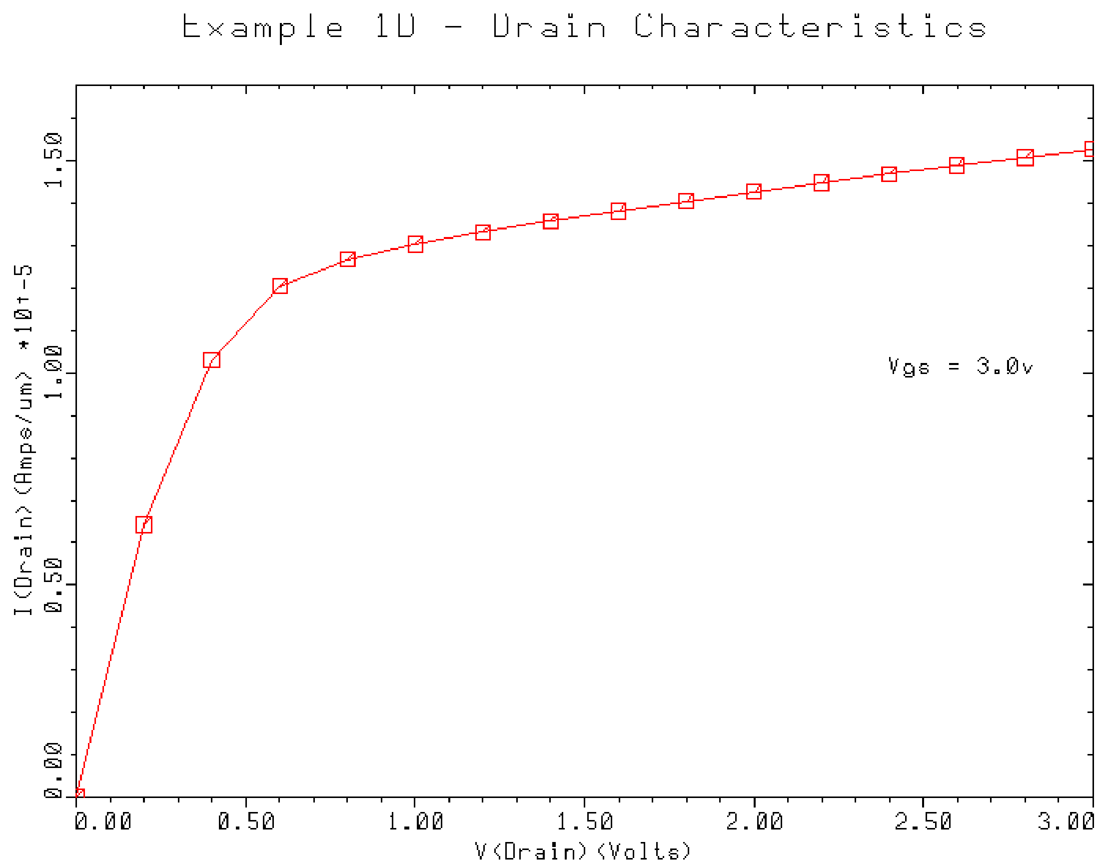
漏极上加上步长为VSTEP，扫描次数为NSTEP的扫描电压，然后进行模拟。

```
COMMENT    Plot Ids vs. Vds
PLOT.1D    Y.AXIS=I(Drain)  X.AXIS=V(Drain)  POINTS  COLOR=2
+          TITLE="Example 1D - Drain Characteristics"
```

该语句显示漏极电压（横坐标）和漏极电流（纵坐标）的关系，结果下图：

```
LABEL      LABEL="Vgs = 3.0v"  X=2.4  Y=0.1E-4
```

LABEL语句用来在图上适当位置添加标志.



```
COMMENT    Potential contour plot using most recent solution
PLOT.2D    BOUND   JUNC   DEPL   FILL   SCALE
+          TITLE="Example 1D - Potential Contours"
E.LINE     X.START=2.3   Y.START=0.02   S.DELTA=-0.3   N.LINES=8
+          LINE.TYPE=3   COLOR=1
```

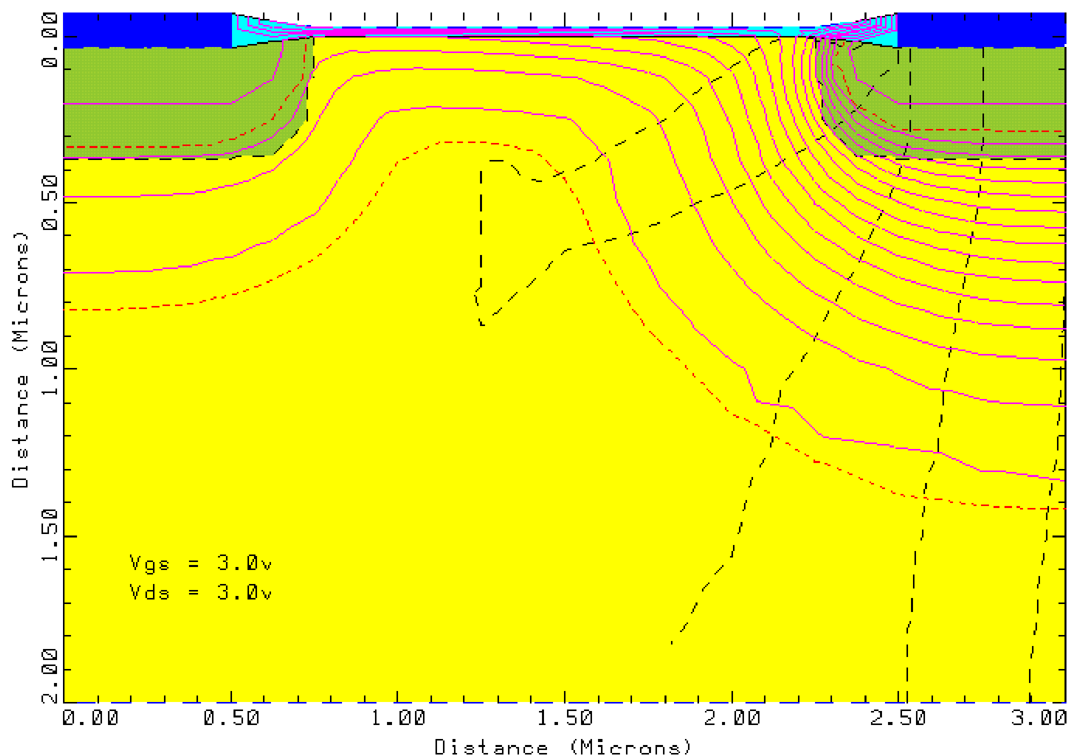
E. LINE 是用来画电力线的,这条语句必须和 PLOT.1D 或者是 PLOT.2D 相结合使用.在这里要求最多画 N.LINES 条电力线,从(X.START,Y.START)开始画，S .DELTA 定义了电力线起点之间的距离,正数表示在上一个条电力线的右边，负数表示在左边。.

```
CONTOUR    POTENTIA   MIN=-1   MAX=4   DEL=.25   COLOR=6
```

这一条语句是用来绘制势能分布的（由参数POTENTIA决定），绘制的势能曲线从-1伏（MIN)开始,到4伏(MAX),每一条曲线之间电势差为0.25伏(DEL),共有(MAX-MIN)/DEL条势能曲线。

```
LABEL      LABEL="Vgs = 3.0v"   X=0.2   Y=1.6
LABEL      LABEL="Vds = 3.0v"
```

这两条语句在图中加了两个标志,使图形更具有可读性。上面几句绘制的势能曲线如下：

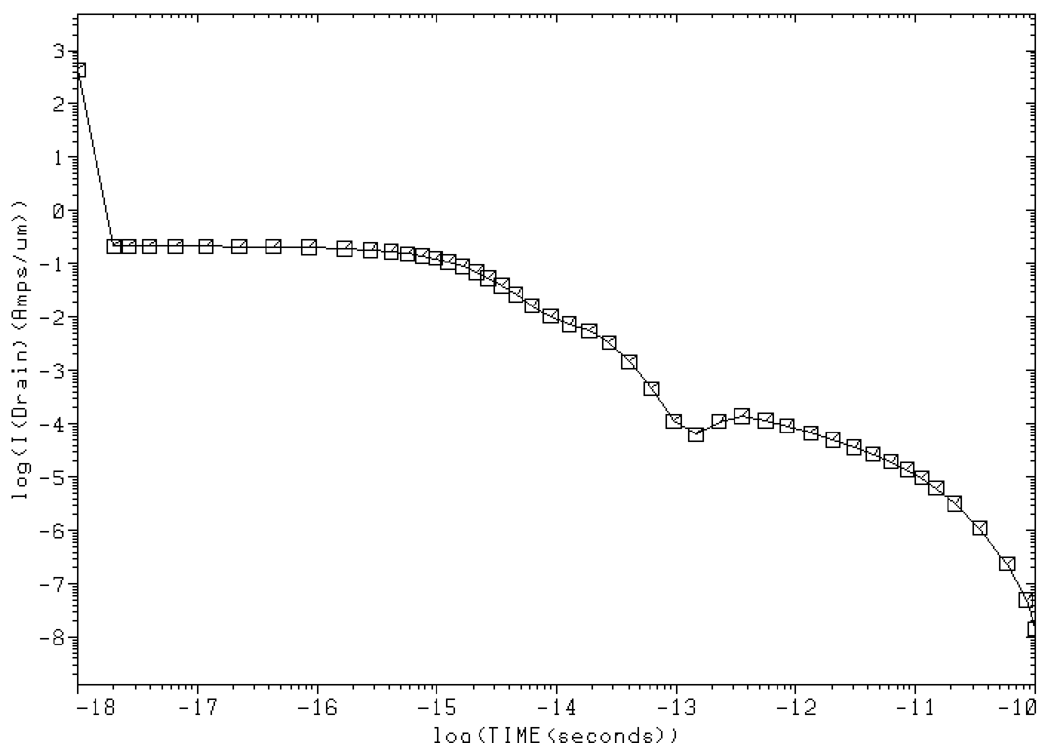



```
SOLVE      V(Drain)=0   TSTEP=1E-18   TSTOP=1E-10
```

下面将要显示当漏极电压突然从 5 伏（上面一个 SOLVE 语句已经得到了）突然降到 0 伏（在这一个 SOLVE 语句中由 V(Drain)得到）时的漏极电流瞬态曲线,因为瞬态响应的模拟不同于直流模拟,因而必须重新求解,在这里,设定求解时迭代的步长为 TSTEP,模拟结束时间为 TSTOP.

```
PLOT.1D    X.AXIS=TIME   Y.AXIS=I(Drain)    Y.LOG   X.LOG POINTS
```

这个语句设定纵坐标为漏极电流,横坐标为时间,两个坐标都使用对数坐标。



§5－4　教学实例 2

下面是一个 npn 双集型三极管的描述实例：

1... TITLE TMA MEDICI Example 2P - NPN Transistor Simulation

2... COMMENT Simulation with Modified Emitter Region

3... COMMENT Initial mesh specification

4... MESH;创建一个原始网格

5... X.MESH WIDTH=6.0 H1=0.250；网格横向宽为6u，间距为0.25u

6... Y.MESH Y.MIN=-0.25 Y.MAX=0.0 N.SPACES=2；在纵向－0.25和0之间创建两（N.SPACES）行网格

7... Y.MESH DEPTH=0.5 H1=0.125；纵向添加深度为0.5u的网格，纵向间距为0.125u

8... Y.MESH DEPTH=1.5 H1=0.125 H2=0.4；纵向再添加深度为1.5u的网格，其纵向间距从0.125u变化到0.4u

9... COMMENT Region definition

10... REGION NAME=Silicon SILICON；定义整个区域性质为silicon

11... REGION NAME=Oxide OXIDE Y.MAX=0；定义从－0.25到0的区域都为二氧化硅

12... REGION NAME=Poly POLYSILI Y.MAX=0 X.MIN=2.75 X.MAX=4.25；再次定义二氧化硅层的中间部分区域为poly

13... COMMENT Electrodes

14... ELECTR NAME=Base X.MIN=1.25 X.MAX=2.00 Y.MAX=0.0；基区电极位置定义

15... ELECTR NAME=Emitter X.MIN=2.75 X.MAX=4.25 TOP；发射区电极位置定义（在整个器件顶部,TOP）

16... ELECTR NAME=Collector BOTTOM；集电区电极位置定义（在器件的最底部BOTTOM）

17... COMMENT Specify impurity profiles

18... PROFILE N-TYPE N.PEAK=5e15 UNIFORM OUT.FILE=MDEX2DS；定义衬底为n型均匀搀杂，浓度为5e15，并将所有定义的搀杂特性记录在文件MDEX2DS中，在下次网格优化时方便调用

19... PROFILE P-TYPE N.PEAK=6e17 Y.MIN=0.35 Y.CHAR=0.16

... + X.MIN=1.25 WIDTH=3.5 XY.RAT=0.75；定义基区为p型搀杂，浓度为6e17，搀杂特征长度（Y.CHAR）为0.16，横向扩散率为0.75

20... PROFILE P-TYPE N.PEAK=4e18 Y.MIN=0.0 Y.CHAR=0.16

... + X.MIN=1.25 WIDTH=3.5 XY.RAT=0.75;仍旧是定义基区的搀杂特性（和发射区邻接部分浓度较高）

21... PROFILE N-TYPE N.PEAK=7e19 Y.MIN=-0.25 DEPTH=0.25 Y.CHAR=0.17

... + X.MIN=2.75 WIDTH=1.5 XY.RAT=0.75；定义n型发射区的搀杂特性

22... PROFILE N-TYPE N.PEAK=1e19 Y.MIN=2.0 Y.CHAR=0.27；定义n型集电区的搀杂特性

23... COMMENT Regrids on doping

24... REGRID DOPING LOG RATIO=3 SMOOTH=1 IN.FILE=MDEX2DS；读入文件MDEX2DS,对网格进行优化处理，当网格上某节点的搀杂变化率超过3时，对这个网格进行更进一步的划分（分为四个全等的小三角形）

25... REGRID DOPING LOG RATIO=3 SMOOTH=1 IN.FILE=MDEX2DS；再次进行同样的优化处理，将网格更加的细化

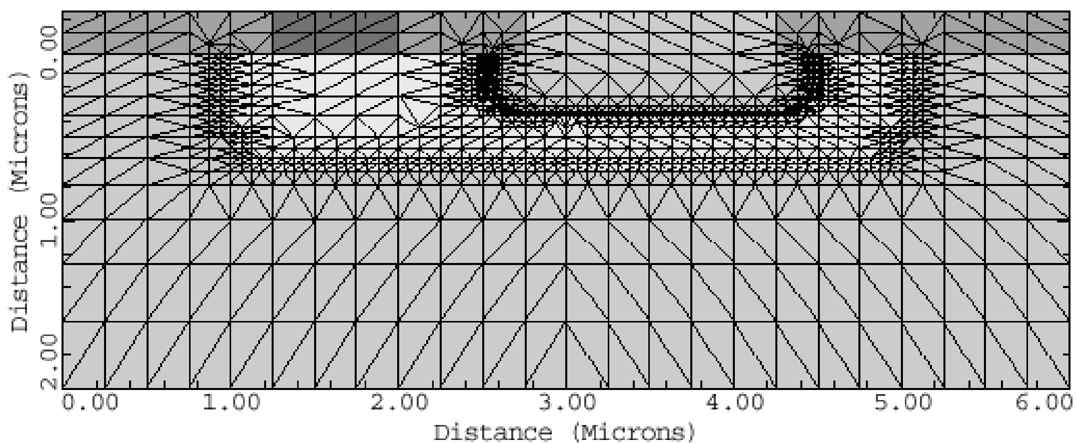
26... COMMENT Extra regrid in emitter-base junction region only.

27... REGRID DOPING LOG RATIO=3 SMOOTH=1 IN.FILE=MDEX2DS

... + X.MIN=2.25 X.MAX=4.75 Y.MAX=0.50 OUT.FILE=MDEX2MP；对发射区与基区交界部分的网格进行专门的优化处理。最后将整个完整定义的网格保存在文件MDEX2MP中

28... PLOT.2D GRID SCALE FILL

... + TITLE="Example 2P - Modified Simulation Mesh"；完成的网格如下图



29... COMMENT Modify properties of polysilicon-emitter region

30... MOBILITY POLYSILI CONC=7E19 HOLE=2.3 FIRST LAST；在多晶硅的搀杂浓度为7e19时，空穴的迁移率为2.3（依赖多晶硅的搀杂浓度而变化），不过FIRST和LAST这两个参数的引入表明无论搀杂浓度为多少，空穴的迁移率保持不变

31... MATERIAL POLYSILI TAUP0=8E-8；多晶硅中空穴的寿命保持为8e－8

32... MODEL CONMOB CONSRH AUGER BGN；定义在模拟中用到的各种物理模型，CONMOB表示使用迁移率与杂质分布有关的模型； AUGER表示使用与俄歇复合有关的模型；BGN表示使用与禁带宽度变窄效应

有关的模型。

33... COMMENT Initial solution

34... SYMB CARRIERS=0；在SYMB语句中如果设置CARRIERS＝0，表示只选用POISSON方程来建模。称之为零载流子模型

35... METHOD ICCG DAMPED；一般使用上述两个参数来解决零载流子模型

36... SOLVE V(Collector)=3.0；在Vc＝3v时求探索解

37... SYMB NEWTON CARRIERS=2；在使用了零载流子模型作初步估计后，我们使用更精确的模型：NEWTON来作进一步求解

38... SOLVE；仍旧在Vc＝3v时求解（使用NEWTON模型）

39... COMMENT Setup log files, forward bias base-emitter junction, and

... + calculate the admittance matrix（导纳矩阵） at 1.0 MHz

40... LOG OUT.FILE=MDEX2PI；将上面模拟的数据保存在LOG文件MDEX2PI中,后面要用到

41... SOLVE V(Base)=0.2 ELEC=Base VSTEP=0.1 NSTEP=4

... + AC.ANAL FREQ=1E6 TERM=Base；在频率为1e6HZ，Vb＝0.2v－0.6v（步长为0.1V）的情况下，进行交流小信号的模拟

42... SOLVE V(Base)=0.7 ELEC=Base VSTEP=0.1 NSTEP=2

... + AC.ANAL FREQ=1E6 TERM=Base OUT.FILE=MDEX2P7；同样是在频率为 1e6HZ，Vb＝0.7－0.9（步长为 0.1V）的情况下，进行交流小信号的模拟，并将结果(Vb＝0.7v)保存在文件 MDEX2P7 中,Vb=0.8v 的结果保存在文件 MDEX2P8 中，Vb=0.9v 的结果保存在文件 MDEX2P9 中

1... TITLE TMA MEDICI Example 2PP - NPN Transistor Simulation

2... COMMENT Post-Processing of MDEX2P Results

3... COMMENT Plot Ic and Ib vs. Vbe

4... PLOT.1D IN.FILE=MDEX2PI Y.AXIS=I(Collector) X.AXIS=V(Base)

... + LINE=1 COLOR=2 TITLE=”Example 2PP - Ic & Ib vs. Vbe”...

+ BOT=1E-14 TOP=1E-3 Y.LOG POINTS；读取LOG文件，绘制集电极电流和基极电压的关系曲线，其中纵坐标为对数坐标（LOG文件一般与PLOT.1D联合使用）。

5... PLOT.1D IN.FILE=MDEX2PI Y.AXIS=I(Base) X.AXIS=V(Base)

... + Y.LOG POINTS LINE=2 COLOR=3 UNCHANGE；绘制基极电流和电压的曲线图，UNCHANGE表明仍旧绘制在上面一条曲线所在的坐标系中。

6... LABEL LABEL=”Ic” X=.525 Y=1E-8

7... LABEL LABEL=”Ib” X=.550 Y=2E-10

8... LABEL LABEL=”Vce = 3.0v” X=.75 Y=1E-13；上述三句在上面绘制的曲线图上添加标签

9... COMMENT Plot the current gain (Beta) vs. collector current

10... EXTRACT NAME=Beta EXPRESS=@I(Collector)/@I(Base)；使用EXTRACT语句，列出Beta（增益）的表达式

11... PLOT.1D IN.FILE=MDEX2PI X.AXIS=I(Collector) Y.AXIS=Beta

... + TITLE=”Example 2PP - Beta vs. Collector Current”

... + BOTTOM=0.0 TOP=25 LEFT=1E-14 RIGHT=1E-3

... + X.LOG POINTS COLOR=2；绘制集电极电流与增益的关系曲线

12... LABEL LABEL=”Vce = 3.0v” X=5E-14 Y=23；做标签

13... COMMENT Plot the cutoff frequency Ft=Gcb/(2*pi*Cbb)

14... EXTRACT NAME=Ft UNITS=Hz

... + EXPRESS=”@G(Collector,Base)/(6.28*@C(Base,Base))”；列出截止频率的表达式，单位是Hz

15... PLOT.1D IN.FILE=MDEX2FI X.AXIS=I(Collector) Y.AXIS=Ft

... + TITLE="Example 2FP - Ft vs. Collector Current"

... + BOTTOM=1 TOP=1E10 LEFT=1E-14 RIGHT=1E-3

... + X.LOG Y.LOG POINTS COLOR=2；绘制集电极电流与截止频率的关系曲线，横纵坐标均使用对数坐标

16... LABEL LABEL="Vce = 3.0v" X=5E-14 Y=1E9；做标签

17... COMMENT Read in the simulation mesh and solution for Vbe=0.9v

18... MESH IN.FILE=MDEX2MS；由于要绘制二维图形，为了方便，重新载入前面描述的网格。

19... LOAD IN.FILE=MDEX2S9；载入模拟结果文件MDEX2S9(Vbe＝0.9v)

20... COMMENT Vector plot of total current for Vbe=0.9v

21... PLOT.2D BOUND JUNC SCALE FILL

... + TITLE="Example 2FP - Total Current Vectors"

22... VECTOR J.TOTAL COLOR=2；绘制二维电流矢量图

23... LABEL LABEL="Vbe = 0.9v" X=0.4 Y=1.55

24... LABEL LABEL="Vce = 3.0v"；做标签

25... COMMENT Potential contour plot for Vbe=0.9v

26... PLOT.2D BOUND JUNC DEPL SCALE FILL

... + TITLE="Example 2FP - Potential Contours"

27... CONTOUR POTEN MIN=-1 MAX=4 DEL=.25 COLOR=6；绘制等势能曲线（CONTOUR用来绘制等高线），POTEN指势能，MIN和MAX指定参数的显示范围，DEL表示所显示的相邻曲线在参数值上的间隔，负数表示是p型掺杂，正数表示是n型掺杂

28... LABEL LABEL="Vbe = 0.9v" X=0.4 Y=1.55

29... LABEL LABEL="Vce = 3.0v"；做标签

30... COMMENT Plot doping and carrier concentrations for Vbe=0.7v

31... LOAD IN.FILE=MDEX2S7；载入模拟结果文件MDEX2S7(Vbe＝0.7v)

32... PLOT.1D DOPING Y.LOG SYMBOL=1 COLOR=2 LINE=1

... + BOT=1E10 TOP=1E20

... + X.STA=3.5 X.END=3.5 Y.STA=0 Y.END=2

... + TITLE="Example 2FP - Carrier & Impurity Conc.";绘制器件的杂质浓度特性曲线，使用第一种标志（SYMBOL=1,方块），起始点为（3.5,0）,终止点为（3.5,2）

33... PLOT.1D ELECTR Y.LOG SYMBOL=2 COLOR=3 LINE=2 UNCHANGE

... + X.STA=3.5 X.END=3.5 Y.STA=0 Y.END=2；仍旧在上面曲线的基础上绘制电子的浓度特性曲线

34... PLOT.1D HOLES Y.LOG SYMBOL=3 COLOR=4 LINE=3 UNCHANGE

... + X.STA=3.5 X.END=3.5 Y.STA=0 Y.END=2；绘制空穴的浓度特性曲线

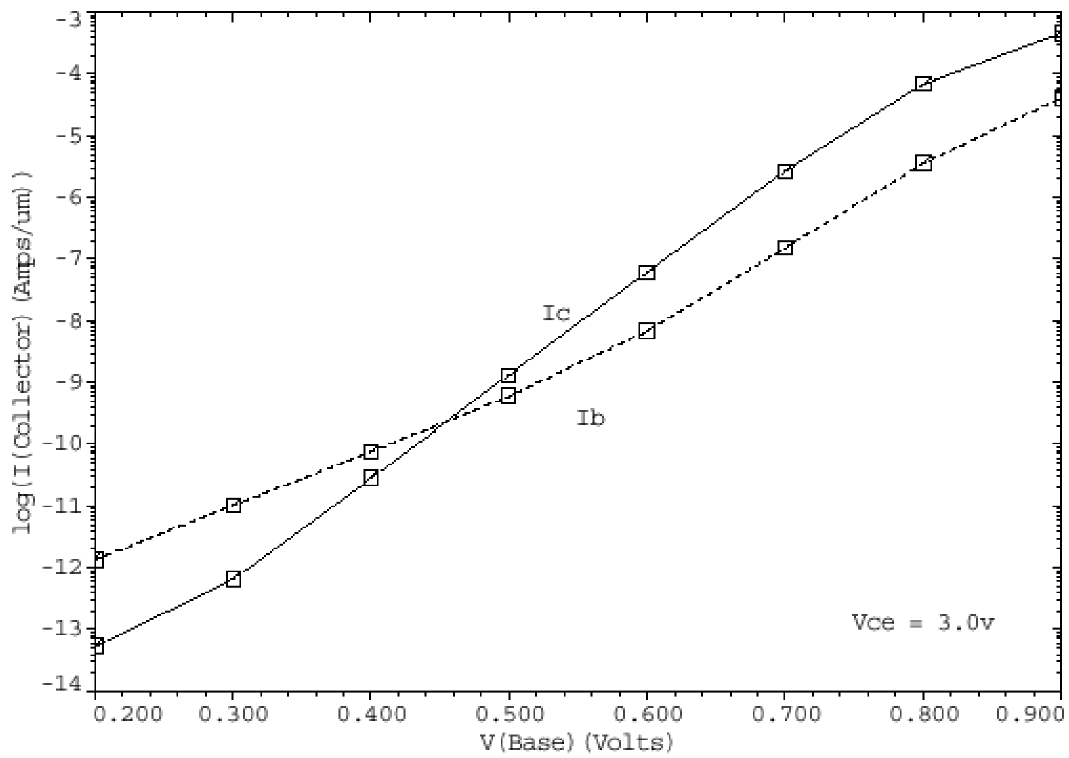
35... LABEL LABEL="Vbe = 0.7v" X=1.55 Y=4E12

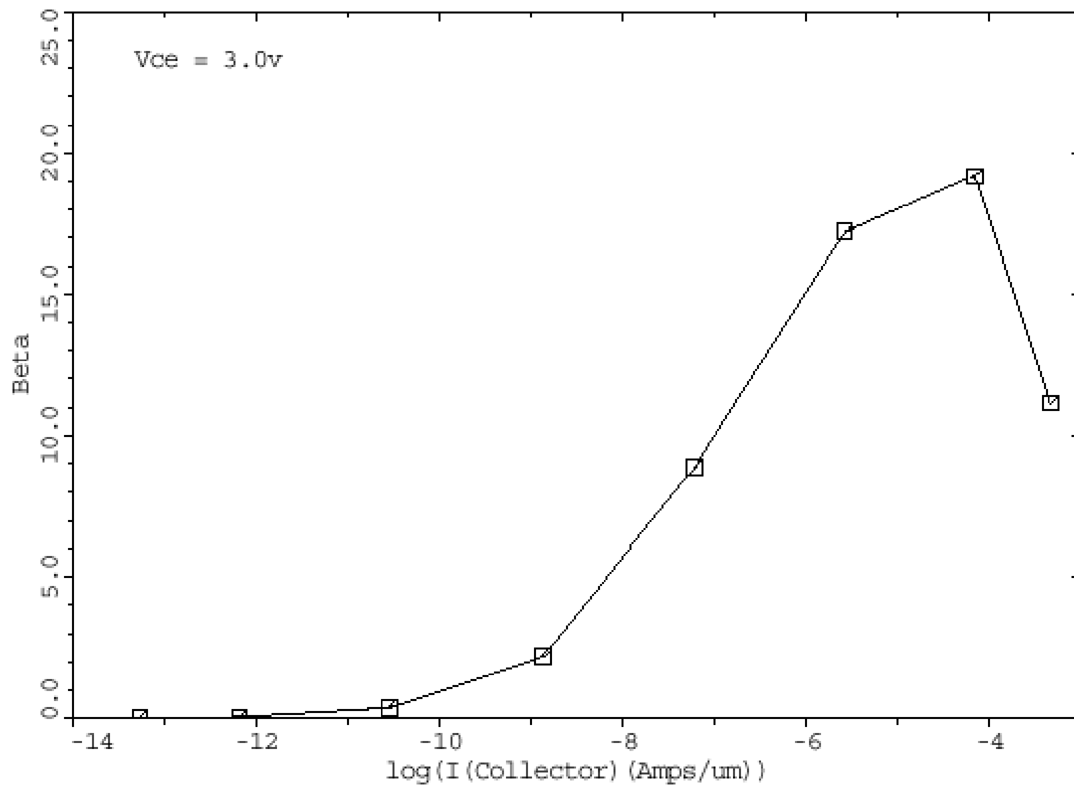
36... LABEL LABEL="Vce = 3.0v"

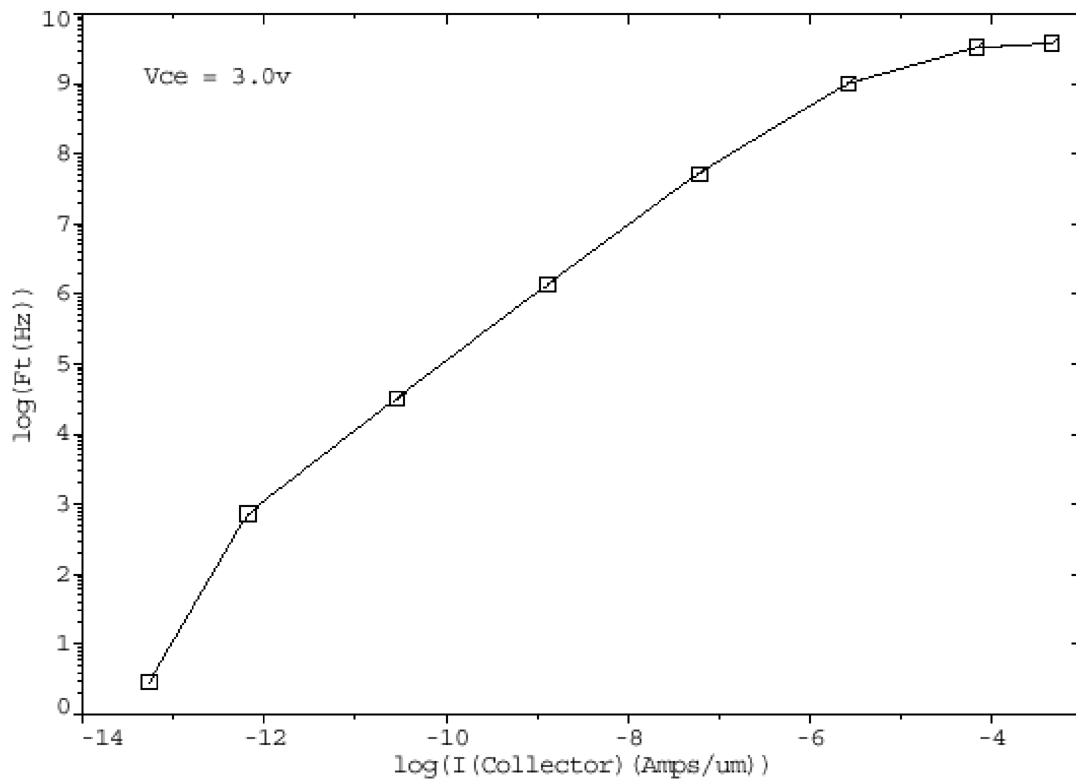
37... LABEL LABEL="Doping" SYMBOL=1 COLOR=2

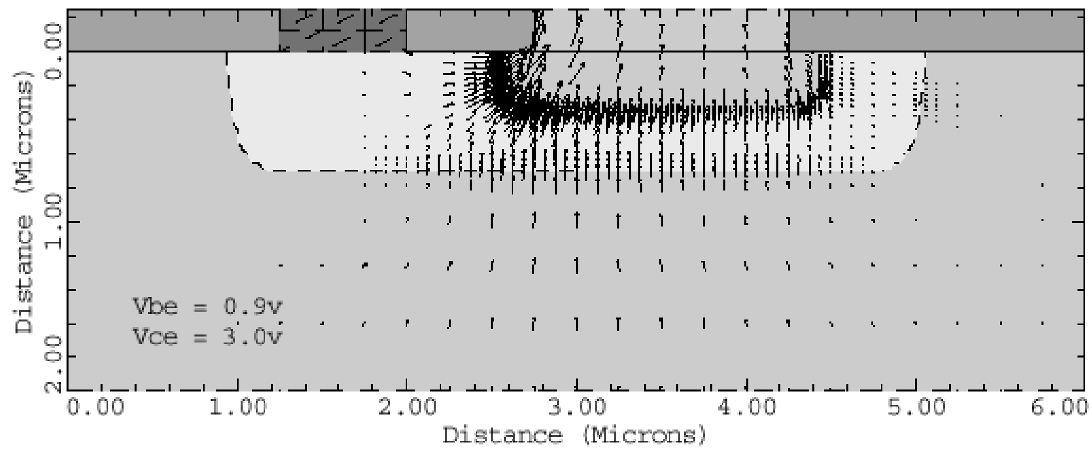
38... LABEL LABEL="Electrons" SYMBOL=2 COLOR=3

39... LABEL LABEL="Holes" SYMBOL=3 COLOR=4；添加标签

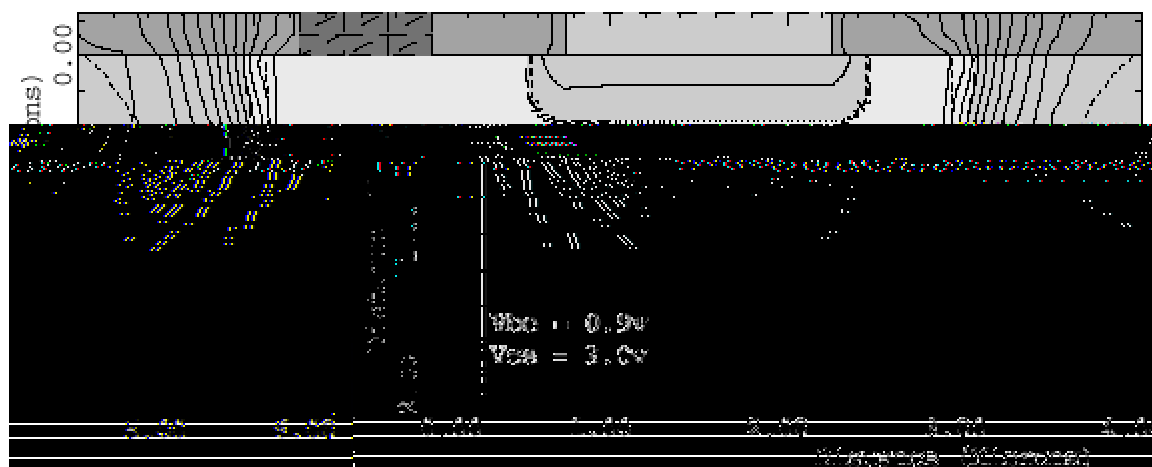

Example 2PP - Ic & Ib vs. Vbe
log(I(Collector)(Amps/um))
V(Base)(Volts)
Ic
Ib
Vce = 3.0v
-3
-4
-5
-6
-7
-8
-9
-10
-11
-12
-13
-14
0.200
0.300
0.400
0.500
0.600
0.700
0.800
0.900


Example 2PP - Beta vs. Collector Current
Beta
log(I(Collector)(Amps/um))
Vce = 3.0v
0.0
5.0
10.0
15.0
20.0
25.0
-14
-12
-10
-8
-6
-4

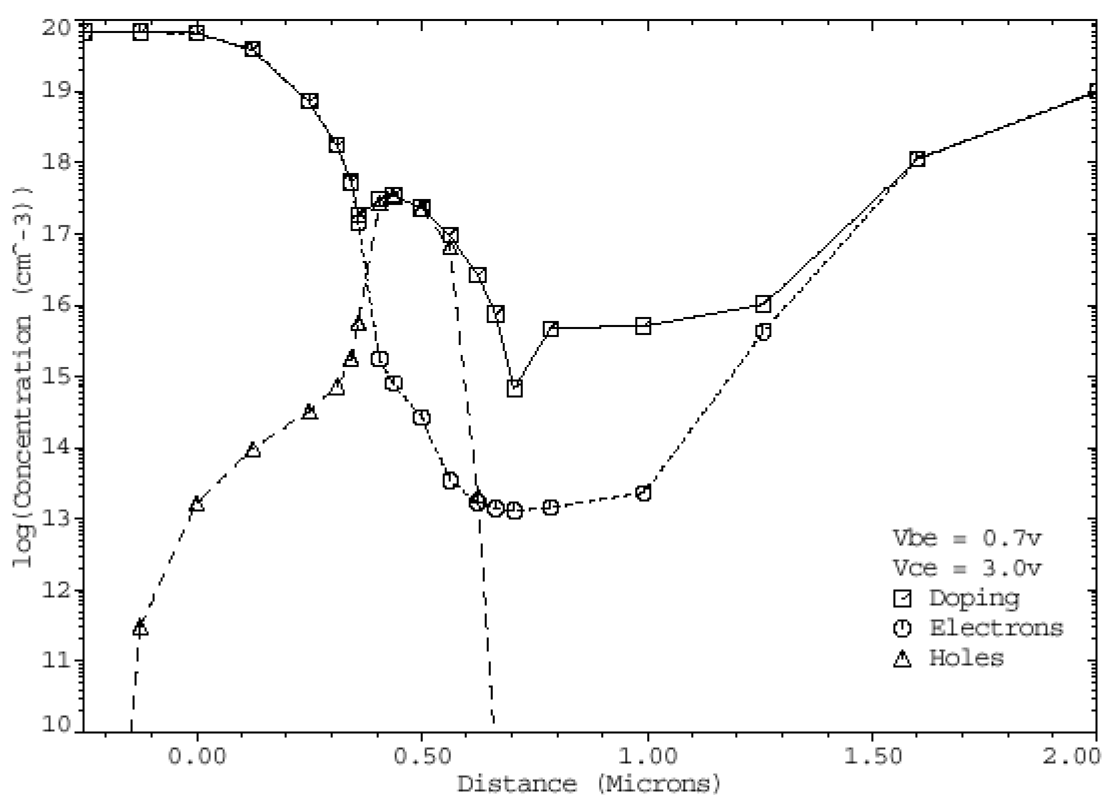

Example 2PP - Ft vs. Collector Current
Vce = 3.0v
log(Ft(Hz))
log(I(Collector)(Amps/um))
0
1
2
3
4
5
6
7
8
9
10
-14
-12
-10
-8
-6
-4


Example 2PP - Total Current Vectors
Vbe = 0.9v
Vce = 3.0v
Distance (Microns)
Distance (Microns)
0.00
1.00
2.00
3.00
4.00
5.00
6.00

Example 2PP - Potential Contours



Example 2PP - Carrier & Impurity Conc.



第六章　　附录

§6－1　前文非门规则文件的编写

附录1：DRC文件

```
drcExtractRules(
        nwell=geomOr("nwell")
        nselect=geomOr("nselect")
        pselect=geomOr("pselect")
        poly=geomOr("poly")
        active=geomOr("active")
        contact=geomOr("contact")
        metal1=geomOr("metal1")
        metal2=geomOr("metal2")
        via=geomOr("via")
        ndiff=geomAnd( active nselect)
        pdiff=geomAnd( active pselect)
        ngate=geomAnd(ndiff poly)
        pgate=geomAnd( pdiff poly)

ivIf(switch("drc?") then
    ivIf(switch("checkTechFile")    then
            checkAllLayers()
      )
;*/rules for nwell
    ivIf(switch("nwell")||switch("all")    then
  drc(nwell width < 4.8 "1.a:Minimum nwell width =4.8")
   drc(nwell sep < 1.8 "1.b:Minimum nwell to nwell spacing =1.8")
   drc(nwell ndiff enc < 0.6 "1.c:nwell enclosure ndiff =0.6" )
  drc(nwell pdiff enc < 1.8 "1.d:nwell enclosure pdiff =1.8")
  saveDerived(geomAndNot(pgate nwell) "1.e:p mos device must in nwell")
)
  ;*/rules for active
    ivIf(switch("active")||switch("all")    then
  drc(active width < 1.2 "2.a:Minimum active width =1.2" )
  drc(active sep < 1.2 "2.b:Minimum active to active spacing =1.2")
)
  ;*/rules for poly
ivIf(switch("poly")||switch("allInterconnect")||switch("all") then
  drc(poly    width < 0.6 "3.a:Minimum poly width<0.6")
drc(poly    sep < 0.6      "3.b:Poly to Poly spacing<0.6")
drc(poly active    sep < 0.6    "3.c:Field Poly to Active spacing <0.6")
    ngatel=geomGetEdge(ngate coincident poly)
    ngatew=geomGetEdge(ngate inside poly)
```

```
    pgatel=geomGetEdge(pgate coincident poly)
    pgatew=geomGetEdge(pgate inside poly)
      drc(poly ngatew
                    enc < 0.6 opposite "3.d:nPoly gate overlap onto field<.6"
              )
              drc(active ngatel
                 enc < 0.6 opposite "3.e:nSource/Drain enclosure of gate<.6"
              )
               drc(poly pgatew
                    enc < 0.6 opposite "3.d:pPoly gate overlap onto field<.6"
              )
              drc(active pgatel
                    enc < 0.6 opposite "3.e:pSource/Drain enclosure of gate<.6"

              )
  )

  ;*/contact rules
  ivIf(switch("contact")||switch("allInterconnect")||switch("all") then
  saveDerived(geomAndNot(contact geomOr(active poly)) "contact not inside Active or
poly")
  saveDerived(geomAndNot(contact metal1) "contact not covered by Metal1")
            drc(contact
                    width < 0.6 "4.a:Contact width<0.6"
            )

            drc(contact
                    sep < 0.9 "4.b:Contact to Contact spacing <0.9"
            )

            drc(poly contact
                    enc < 0.3 "4.c:Contact inside Poly<0.3"
            )
            saveDerived(geomStraddle(contact    poly)
                   "4.c:contact inside poly<0.3")
            drc(metal1 contact
                     enc < 0.3 "4.d:Contact inside Metal1<0.3"
            )
             saveDerived(geomStraddle(contact    metal1)
                   "4.d:contact inside metal1<0.3")

             saveDerived(geomOutside(contact metal1)
                   "4.d:contact inside metal1<.3")
   )
```

```
;*/metal1 rules
ivIf(switch("metal1")||switch("allInterconnect")||switch("all") then
          drc(metal1
                   width < 1.2 "5.a:Metal1 width<1.2"
           )
          drc(metal1
                   sep < 0.9 "5.b:Metal1 to Metal1 spacing <0.9"
           )
)

;*/metal2 rules
ivIf(switch("metal2")||switch("allInterconnect")||switch("all") then
          drc(metal2
                  width < 1.2 "6.a:Metal2 width<1.2"
         )
         drc(metal2
                  sep < 1.2 "6.b:Metal2 to Metal2 spacing<1.2"
         )
         drc(metal2
                  notch < 1.2 "6.c:Metal2 to Metal2 spacing<1.2"
         )
  )

;*/via rules
ivIf(switch("via")||switch("allInterconnect")||switch("all") then
         drc(via
                   width < 0.6 "7.a:Via width<0.6"
          )
         drc(via
                     sep < 0.9 "7.b:Via to Via spacing<0.9"
          )
          drc(via contact
                     sep < 0.6 "7.c:Via to Contact spacing <0.6"
          )
           drc(metal1 via
                     enc < 0.3 "7.d:Via inside Metal1<0.3"
           )
           drc(metal2 via
                     enc < 0.3 "7.e:Via inside Metal2<0.3"
           )
 saveDerived(geomAndNot(via metal1) "Via not inside Metal1")
saveDerived(geomAndNot(via metal2) "Via not inside Metal2")
saveDerived(geomOverlap(via contact) "Via not allowed over contacts")
```

```
        saveDerived(geomOverlap(via poly) "Via not allowed over Poly")
                        drc(via poly
                                    sep < 0.3 "7.f:Via to Poly spacing <0.3"
                        )
         )
           )
         )
```

附录 2：extract 文件

```
;/*EXTRACT RULES FOR NWELL CMOS

drcExtractRules(
ivIf( switch( "extract?" ) then
;define commonly used layers
    bkgnd=geomBkgnd()
    nwell=geomOr("nwell")
    psub=geomAndNot(bkgnd nwell)
    active=geomOr("active")
    nselect=geomOr("nselect")
    pselect=geomOr("pselect")
    poly=geomOr("poly")
    contact=geomOr("contact")
    metal1=geomOr("metal1")
    metal2=geomOr("metal2")
    via=geomOr("via")
    ndiff=geomAnd(active nselect)
    pdiff=geomAnd(active pselect)
   ; nactive=geomAnd(ndiff psub)
    ;pactive=geomAnd(pdiff nwell)

 ;define recognition layers
    ngate=geomAnd(ndiff poly)
    pgate=geomAnd(pdiff poly)

  ;define terminal layers
    nsd=geomAndNot(ndiff poly)
    psd=geomAndNot(pdiff poly)
   ;define persudo layers
    ntap=geomAnd(nsd nwell)
    ptap=geomAnd(psd psub)
  geomConnect(
    via(via metal1 metal2 )
    via(contact metal1 psd nsd poly )
    via(ntap nwell nsd)
    via(ptap psub psd)
```

```
)
    ;subPConn=geomStamp( psub ptap error)
      ; subNConn=geomStamp( nwell ntap error)

extractDevice( pgate poly("G") psd("S" "D")    "pmos ivpcell" )
extractDevice( ngate poly("G") nsd("S" "D")    "nmos ivpcell")

;device measurement
pgateWidth = measureParameter( length (pgate coincident poly ) 0.5)
pgateLength = measureParameter( length (pgate inside poly ) 0.5)
saveParameter( pgateWidth "w" )
saveParameter( pgateLength "l" )
ngateWidth = measureParameter( length ( ngate coincident poly ) 0.5 )
ngateLength = measureParameter( length (ngate inside poly ) 0.5 )
saveParameter( ngateWidth "w" )
saveParameter( ngateLength "l" )

;saveInterconnect(contact poly metal1 metal2 via )
saveInterconnect( poly contact metal1 metal2 via)
;saveInterconnect((nsd "nselect") (psd "pselect")    )
;saveInterconnect((subNConn "nwell") (subPConn "pselect"))
saveRecognition( ngate "poly")
saveRecognition( pgate "poly")
)
)
```

附录 3：LVS 文件

```
lvsRules(
procedure( compareMOS( layPlist,schPlist)

prog( ( )
      if(layPlist->w!=nil && schPlist->w!=nil then
                if( layPlist->w !=schPlist->w then
                sprintf(errorW,
                    "Gate width mismatch: %gu layout to %gu schematic",
                          float( layPlist->w ), float( schPlist->w ) )
                      return( errorW )
                )
              )
        if( layPlist->l !=nil && schPlist->l !=nil then
            if( layPlist->l != schPlist-> then
                  sprintf( errorL,
                      "Gate length mismatch: %gu layout to %gu schematic",
                            float( layPlist->l ),float(schPlist->l) )
                  return( errorL )
```

```
        )
       )
      return( nil )
     ) ; prog
    ); comparemos


procedure( parallelMOS( m1Plist, m2Plist )

prog( ( parMos )
  (parMos = (ncons nil ) )
    if(( m1Plist->l !=nil && m2Plist->l !=nil) then
      parMos->l = ( m1Plist->l + m2Plist-> l ) /2.0
      )
    if(( m1Plist->w != nil && m2Plist->w != nil) then
      parMos->w = ( m1Plist->w + m2Plist->w )
      )

   return( parMos )
    ) ; prog
    ) ; parallelMOS
permuteDevice( parallel "pmos" parallelMOS )

permuteDevice( parallel "nmos" parallelMOS )
;permuteDevice( MOS "pmos" )

;permuteDevice( MOS "nmos" )
compareDeviceProperty( "pmos" compareMOS )

compareDeviceProperty( "nmos" compareMOS )
)
;lvsRules
```

§6－2　一个完整DIVA文件的注解

一个完整 DIVA 文件的注解

;*************DRC　Procedure****************

；以下这段将会产生覆盖原焊点（pad）4u 的金属 2 的新层次

drcExtractRules(

  ivIf(switch(“padFix”)||switch(“all”)　then

    metal2=geomOr(“metal2”　geomSize(“pad” 4.0))；将原金属 2 层次中覆盖焊点的部分扩展 4u

```
      geomErase(“metal2”)
      saveDerived(metal2(“metal2”  “drawing”) )；以上两句联合使用将旧层次用
    新层次覆盖，并存入 drawing（原始版图的图层）中
)
；如果通孔（via）在 poly1 上或者在它的 0.5u 范围以内，就产生新层次 viaE
；viaE 图形必须覆盖通孔（via）0.2u
ivIf(switch(“viaEfix”)||switch(“all”)  then
   viaToFix=geomOverlap(“via”  geomSize(“poly1”  0.5))；使用这条语句选择满
   足条件的 via（先将 poly1 扩展 0.5u，然后选择与新图形相交的 via）
   viaE=geomSize(viaToFix  0.2)；产生 viaE 层，它覆盖满足条件的 via0.2u
   geomErase(“viaE”)
   saveDerived(viaE  (“viaE”  “drawing”));将新产生的 viaE 层存入 drawing 中
)
)
;*************DRC   Procedure****************
drcExtractRules(
；生成常用的衍生层
     nsd=geomAndNot(“ndiff”   “poly1”)
     psd=geomAndNot(“pdiff”   “poly1”)
     ngate=geomAnd(“ndiff”    “poly1”)
     pgate=geomAnd(“pdiff”    “poly1”)
；开始 DRC 校检
     ivIf(switch(“drc?”)  then
         dummy=geomBkgnd()
；进行 0 尺寸图形检查
         ivIf(switch(“0width”)   then
           layers=geomcat(“thinox”    “metal1”   “poly1”   “metal3”   “via2”   “via”
            “contact”  “ndiff”  “PRES”  “pdiff”  “nwell”  “buried”  “pbase”  “pad”)；注
            意 geomCat 和 geomOr 的区别
     drc(layers  width==0  raw  “0-width  shape”)；raw 表明检查的是原始层次。
         )
         ivIf(switch(“offgrid”)   then；进行掩膜错位检查
            maskedLayers=geomCat(“thinox”  “metal1”  “poly1”  “metal3”  “via2”  “via”
            “contact”  “ndiff”  “PRES”  “pdiff”  “nwell”  “buried”  “pbase”  “pad”)
     offGrid(maskedLayers  .05  raw  “shape  off .05   grid”)；使用 offGrid 这个命令来进
            行检查
      )
     ；以下这段将对所有层次进行规定的检查
      ivIf(switch(“checkTechFile”)  then
 checkAllLayers()
 )
   ；阱的规则检查
 ivIf(switch(“well”)||switch(“all”)  then
     drc(“nwell”  width<1.8   “1a:minimum  nwell width =1.8”)；n 阱最小宽度为 1.8u
```

```
drc(“nwell” sep<2.5 “1b:minimum nwell spacing=2.5”)；n 阱间最小间距为 2.5u
drc(“nwell” “pdiff” enc<2.0 “1c:nwell enclosure of pdiff =2.0”)；pdiff 的外边界到 nwell 的内边界距离检查，最小间距为 2u
saveDerived( geomStraddle(“pdiff” “nwell”) “1c:nwell enclosure of pdiff=2.0”)；pdiff 和 nwell 仅有部分交叠，则输出错误信息
drc(“nwell” “ndiff” enc<1.0 “1d:nwell enclosure of ndiff =1.0”)；ndiff 的外边界到 nwell 内边界的最小间距为 1u
saveDerived( geomStraddle(“ndiff” “nwell”) “1d:nwell enclosure of ndiff=1.0”)；ndiff 和 nwell 仅有部分交叠，则输出错误信息

；soft connect 检查
ptap=geomOverlap(“ndiff” “nwell”)；选择在 nwell 上的 ndiff
nwell=geomOr(“nwell”)
ndiff=geomOr(“ndiff”)
metal1=geomOr(“metal1”)
geomConnect(
  via(ptap ndiff metal1)；ndiff 通过 ptap 和 metal1 相连
)
ivIf(switch(“currentCell?”)||!switch(“hier?”)；只在单元内检查阱的连接
    geomStamp(nwell ndiff error)
)
)
；扩散区的规则检查
ivIf(switch(“diff”)||switch(“all”) then
  drc(“ndiff” width<1.0 “2.a:minimum ndiff width=1.0”);ndiff 的最小宽度为 1u
  drc(“pdiff” width<1.0 “2.a:minimum pdiff width=1.0”)；pdiff 的最小宽度为 1u
  drc(“ndiff” sep<1.0 “2.b:minimum ndiff spacing=1.0”)；ndiff 的间距最小为 1u
  drc(“pdiff” sep<1.0 “2.b:minimum pdiff spacing=1.0”)；pdiff 的间距最小为 1u
  drc(“pdiff” “ndiff” sep<1.0 “2.c:minimum ndiff to pdiff spacing =1.0”)；ndiff 和 pdiff 间的距离最小为 1u
  drc(nsd width<1.0 “2d:minimum source/drain diffusion width =1.0”)
  drc(psd width<1.0 “2d:minimum source/drain diffusion width =1.0”)；源区和漏区的最小宽度均为 1u
  saveDerived(geomAndNot(pgate “nwell”) “2.e:P mos device must be inside nwell”)；将不在 nwell 中的 pmos 输出至错误层
)
；poly1 的规则检查

ivIf(switch(“poly1”)||switch(“allInterconnect”)||switch(“all”) then
  drc(“poly1” width<0.6 “3.a:minimum poly1 width=0.6”)；poly1 的最小宽度为 0.6u
  drc(“poly1” sep<1.0 “3.b:minimum poly1 spacing=1.0”)；poly1 的最小间距为 1u
  drc(“poly1” “ndiff” enc <0.4 “3.c:poly1 extension past ndiff =0.4”)
  drc(“poly1” “pdiff” enc <0.4 “3.c:poly1 extension past pdiff =0.4”)；poly1 跨 ndiff 和 pdiff 的最小距离为 0.4u
```

saveDerived(geomButting(pgate psd keep<2) “3c:poly1 extension past pdiff =0.4”);这里的 keep<2 是指如果和 pgate 相外切的 psd 部分少于两部分（作为一个 mos 管，必须有源和漏，因此和 pgate 相外切且分离的 psd 层必须为两个），就输出错误

saveDerived(geomButting(ngate nsd keep<2) “3c:poly1 extension past ndiff =0.4”)

)

；metal1 规则检查

ivIf(switch(“metal1”)||switch(“allInterconnect”)||switch(“all”) then

drc(“metal1” width<0.8 “4a:minimum metal1 width =0.8”)；metal1 的最小宽度为 0.8

drc(“metal1” sep<1.0 “4b:minimum metal1 spacing =1.0”)；metal1 的最小间距为 1u

)

；contact 规则检查

ivIf(switch(“contact”)||switch(“allInterconnect”)||switch(“all”) then

drc(“contact” width<1.0 “5a:minimum contact width =1.0”)；contact 的最小宽度为 1u

drc(“contact” sep<1.0 “5b:minimum contact spacing =1.0”)；contact 间的最小间距为 1u

drc(“ndiff” “contact” enc<1.0 “5c:ndiff enclosure of contact =1.0”)；ndiff 包含 contact 最小为 1u

saveDerived(geomStraddle(“contact” “ndiff”) “5c:ndiff enclosure of contact =1.0”)；若 contact 与 ndiff 仅有部分交叠，输出错误

drc(“pdiff” “contact” enc<1.0 “5c:pdiff enclosure of contact =1.0”) ；pdiff 包含 contact 最小为 1u

saveDerived(geomStraddle(“contact” “pdiff”) “5c:pdiff enclosure of contact =1.0”) ；若 contact 与 pdiff 仅有部分交叠，输出错误

drc(“metal1” “contact” enc<0.5 “5d:metal1 enclosure of contact =0.5”)；metal1 包含 contact 最小为 0.5u

saveDerived(geomStraddle(“contact” “metal1”) “5d:metal1 enclosure of contact =0.5”) ；若 contact 与 metal1 仅有部分交叠，输出错误

saveDerived(geomOutside(“contact” “metal1”) “5d:metal1 enclosure of contact =0.5”)；若有 contact 在 metal1 之外，输出错误

drc(“poly1” “contact” enc<0.5 “5e:poly1 enclosure of contact =0.5”)；poly1 包含 contact 最小为 0.5u

saveDerived(geomStraddle(“contact” “poly1”) “5e:poly1 enclosure of contact =0.5”)；若 contact 与 poly1 仅有部分交叠，输出错误

savederived(geomOutside(“contact” geomOr(geomOr(“ndiff” “pbase”) geomOr(“pdiff” “poly1”))) “contact must be enclosed by pdiff or ndiff or poly1”)；contact 必须处在 pdiff、ndiff 或是 poly1 中间

)

ivIf(switch(“metal2”)||switch(“allInterconnect”)||switch(“all”) then

drc(“metal2” width<1.0 “6a:minimum metal2 width =1.0”)；metal2 的最小宽度为 1u

drc(“metal2” sep<1.0 “6b:minimum metal2 spacing =1.0”)；metal2 间的最小间距为 1u

)

ivIf(switch(“via”)||switch(“allInterconnect”)||switch(“all”) then

drc(“via” width<1.0 “7a:minimum via width =1.0”)；via 的最小宽度为 1u

drc(“via” sep<1.0 “7b:minimum via spacing=1.0”)；via 间的最小间距为 1u

drc(“metal1” “via” enc<1.0 “7c:metal1 enclosure of via =1.0”)；metal1 包含 via 最小为 1u

saveDerived(geomStraddle(“via” “metal1”) “7c:metal1 enclosure of via =1.0”)；via 同 metal1 仅有部分交叠，输出错误

saveDerived(geomOutside(“via” “metal1”) “7c:metal1 enclosure of via =1.0”) ;若有处于 metal1 之外的 via，输出错误

drc(“metal2” “via” enc<0.5 “7d:metal2 enclosure of via =0.5”)；metal2 包含 via 最小为 1u

saveDerived(geomStraddle(“via” “metal2”) “7d:metal2 enclosure of via =0.5”) ；via 同 metal1 仅有部分交叠，输出错误

saveDerived(geomOutside(“via” “metal2”) “7d:metal2 enclosure of via =0.5”) ；若有处于 metal1 之外的 via，输出错误

)

ivIf(switch(“pad”)||switch(“all”) then

drc(“pad” width<20.0 “8a:minimum pad width =20.0”)；pad 的最小宽度为 20u

drc(“metal2” “pad” enc<3.0 “8b:metal2 enclosure of pad =3.0”)；metal2 包含 pad 最小为 3u

saveDerived(geomStraddle(“pad” “metal2”) “8b:metal2 enclosure of pad =3.0”)；若 metal2 与 pad 仅有部分交叠，输出错误

saveDerived(geomOutside(“pad” “metal2”) “8b:metal2 enclosure of pad =3.0”)；若有处于 metal2 之外的 pad，输出错误

drc(“pad” sep<25.0 “8c:minimum pad spacing =25.0”)；pad 间的最小间距为 25u

drc(“pad” geomGetByLayer(“metal2” “pad” 6) sep<5.0 “8d:minimum metal2 to pad spacing =5.0”)

)；将 pad 扩张 6u，选择与其相交的 metal2，并检测所选的 metal2 与 pad 的间距最小是否为 5u

ivIf(switch(“res”)||switch(“all”) then

drc(“PRES” width<1.0 “9a:minimum PRES width =1.0”)；PRES 的最小宽度为 1u

drc(“PRES” sep<1.0 “9b:minimum PRES spacing =1.0”)；PRES 间的最小间距为 1u

drc(“PRES” “ndiff” enc<0.5 “9c:PRES enclosure of ndiff =0.5”)；PRES 包含 ndiff 最小为 0.5u

saveDerived(geomStraddle(“ndiff” “PRES”) “9c:PRES enclosure of ndiff =0.5”)；若 ndiff 与 PRES 仅部分交叠，输出错误

)

ivIf(switch(“cap”)||switch(“all”) then

drc(“thinox” width<1.0 “10a:minimum thinox width =1.0”)；thinox 的最小宽度为 1u

drc(“thinox” sep<2.0 “10b:minimum thinox spacing =2.0”)；thinox 间的最小间距为 2u

drc(“poly1” “thinox” enc<0.4 “10c:poly1 enclosure of thinox =0.4”)；poly1 包含 thinox 最小为 0.4u

saveDerived(geomStraddle(“thinox” “poly1”) “10c:poly1 enclosure of thinox=0.4”) ;若 thinox 与 poly1 仅部分相交，输出错误

)

ivIf(switch(“bipolar”)||switch(“all”) then

drc(“buried” “pdiff” enc<0.5 “11a:buried enclosure of pdiff =0.5”)；buried 包含

pdiff 最小为 0.5u
saveDerived(geomStraddle(“pdiff” “buried”) “11a:buried enclosure of pdiff=0.5”) 若 pdiff 与 buried 仅有部分交叠，则输出错误
drc(“buried” “ndiff” enc<0.5 “11a:buried enclosure of ndiff =0.5”)；buried 包含 ndiff 最小为 0.5u
saveDerived(geomStraddle(“ndiff” “buried”) “11a:buried enclosure of ndiff=0.5”) 若 ndiff 与 buried 仅有部分交叠，则输出错误
drc(“buried” “pbase” enc<0.5 “11a:buried enclosure of pbase =0.5”)；buried 包含 pbase 最小为 0.5u
saveDerived(geomStraddle(“pbase” “buried”) “11a:buried enclosure of pbase=0.5”)；若 pbase 与 buried 仅有部分交叠，则输出错误
drc(“pbase” “ndiff” enc<0.5 “11b:pbase enclosure of ndiff =0.5”)；pbase 包含 ndiff 最小为 0.5u
saveDerived(geomStraddle(“ndiff” “pbase”) “11b:pbase enclosure of ndiff =0.5”)；若 ndiff 与 pbase 仅有部分交叠，则输出错误
)
)
)
；**********************Extraction Procedure************************
drcExtractRules(
ivIf(switch(“extract?”) then;开始 extraction
;产生一些常用的衍生层次
poly1=geomOr(“poly1”)
pdiff=geomOr(“pdiff”)
ndiff=geomOr(“nwell”)
pbase=geomOr(“pbase”)
buried=geomOr(“buried”)
thinox=geomOr(“thinox”)
PRES=geomOr(“PRES”)
contact=geomOr(“contact”)
metal1=geomOr(“metal1”)
via=geomOr(“via”)
metal2=geomOe(“metal2”)
pad=geomOr(“pad”)
sub=geomBkgnd()

；定义识别层
ngate=geomAnd(ndiff poly1)
pgate=geomAnd(pdiff poly1)
npn=geomAnd(ndiff pbase)；npn 的发射极
pnp=geomHoles(pdiff)；pnp 的发射极（pnp 的集电极包围其发射极，横向 pnp 管）

;定义端口层
nsd=geomAndNot(ndiff poly1)

```
psd=geomAndNot(pdiff poly1)
pnpEmit=geomAnd(psd pnp)；pnp 的发射极
psd=geomAndNot(psd pnpEmit)；重新定义 psd（去除了 pnp 的发射极部分）
resTerm=geomAnd(geomsize(contact 1)    geomAnd(PRES nsd))
PRES=geomAndNot(goemAnd(PRES nsd) resTerm)
nsd=geomAndNot(nsd geomOr(PRES resTerm))

;定义伪接触层
ptap=geomAndNot(geomAndNot(psd nwell) buried)
psd=geomAndNot(psd ptap)
nburied=geomAndNot(geomAnd(nsd buried) pbase)
pbaseConn=geomAndNot(geomAnd(contact pbase) nsd)
contact=geomAndNot(contact pbaseConn)
padContact=geomAnd(metal2 pad)

;define geomConnect statement
geomConnect(via(contact    metal1 nsd psd poly1 ptap pnpEmit resTerm)
                via(via metal1 metal2)
                via(nburied nsd buried)
                via(pbaseConn pbase metal1)
                via(padContact metal2 pad)
               )
nwellConn=geomStamp(nwell nsd)
subConn=geomStamp(sub ptap)

;define extractDevice statemetnt
extractDevice(pgate poly1("G") psd("S" "D") nwellConn("B") "pmos ivpcell")
extractDevice(ngate poly1("G") nsd("S" "D") subConn("B") "nmos ivpcell")
extractDevice(npn nsd("E") pbase("B") buried("C") "npn ivpcell")
extractDevice(pnp pnpEmit("E") psd("C") buried("B") "pnp ivpcell")
extractDevice(thinox metal1("PLUS") poly1("MINUS") "cap ivpcell")
extractDevice(PRES resTerm("PLUS" "MINUS") "res ivpcell")

;measure device sizes and other parameters

;for nmos device:
wn=measureParameter(length (ngate butting nsd)    .5e-6)
ln=measureParameter(length (ngate inside poly1)    .5e-6)
saveParameter(wn "w")
saveParameter(ln "l")

;for pmos device:
wp=measureParameter(length (pgate butting psd)    .5e-6)
lp=measureParameter(length (pgate inside poly1)    .5e-6)
```

```
saveParameter(wp "w")
saveParameter(lp "l")

;for pnp device:
pnpea=measureParameter(area (pnp over pnpEmit) 1e-6)
saveParameter(pnpea "area")

;for npn device:
npnea=measureParameter(area npn 1e-6)
saveParameter(npnea "area")

;for cap
cap=measureParameter(area(thinox) 5e-15)
saveParameter(cap "c")

;for res
wr=measureParameter(length(PRES butting resTerm)    .5)
lr=measureParameter(length(PRES outside resTerm)     .5)
bendsr=measureParameter(bends_all (PRES outside resTerm))
res=calculateParameter(((lr/wr) –(.46*bendsr))*100.0)
saveParameter(res "r")

;output data to extracted layout
saveRecognition(pgate "poly1")
saveRecognition(ngate "poly1")
saveRecognition(npn "ndiff")
saveRecognition(pnp "buried")
saveRecognition(thinox "thinox")
saveRecognition(PRES "PRES")
saveInterconnect(contact metal1 poly1 metal2 via)
saveInterconnect((nsd "ndiff") (psd "pdiff"))
saveInterconnect((nwellConn "nwell") pbase buried)
saveInterconnect((pnpEmit "pdiff") (pbaseConn "contact"))
saveInterconnect((resTerm "PRES") pad)
copyGragphics(("text" "drawing" )    all)
);end extract?
):end drcExtractRules
```

§ 6－3　规则文件中一些定义和关键词的图文解释

层处理

层处理要求创建新的层次。你可以在初始设计中使用它，也可以在随后的验证语句中用到。

输入输出层次

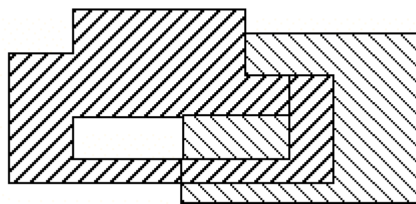
层处理创建的新层次我们称之为衍生层。他可以是多边形（图形）或是边。层处理表达式中的输入层可以是多边形、边、衍生层或是原始的图形层次（使用在真正物理版图设计上的层次）

在层次处理之前，diva 默认将所有在同一层次上的图形合并。有必要的话可以使用 raw 这个命令来使所需的一些层次不合并（在下面会再次提到这个命令）。

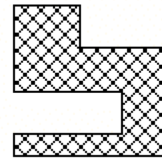
逻辑命令

逻辑命令可以被称作层间的 " 布尔运算 "。他们从输入层出发，创建新的几何图形。不同的命令所需的输入层数目从一到多不等。

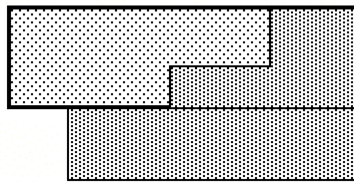
GeomAnd　输出两个不同层次或边界间的交叠部分。一般需要一个或两个输入层。



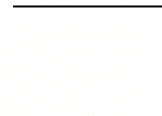
Original polygons



geomAnd result

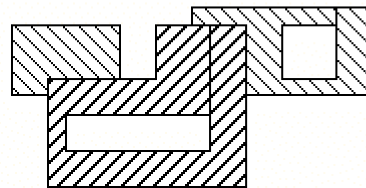


Original edges shown as polygons

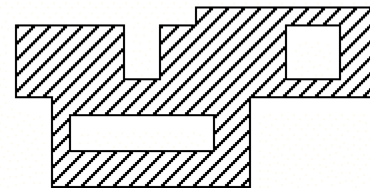


geomAnd result

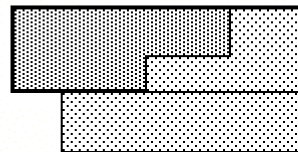
GeomOr　输出所有的输入层。这些层次（边界）将会被合并成为一个新层次。



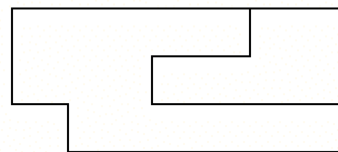
Original polygons



geomOr result

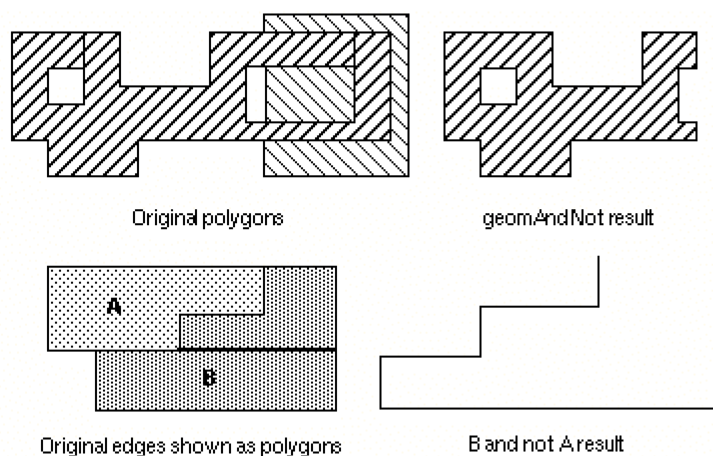


Original edges shown as polygons

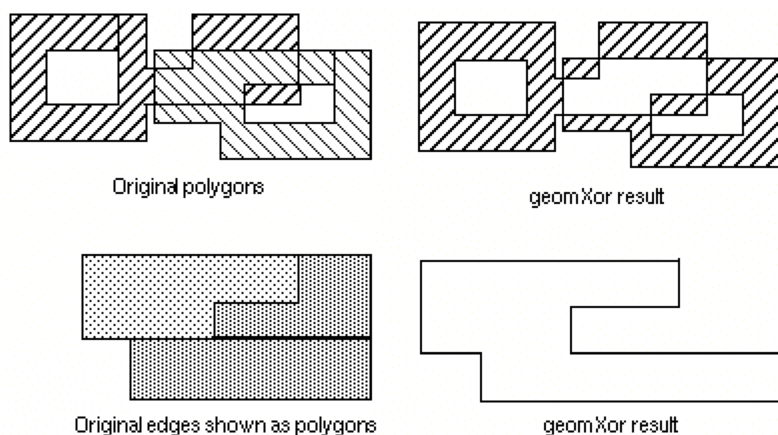


geomOr result

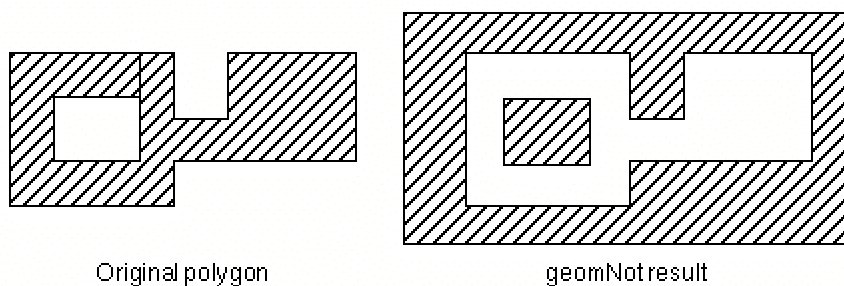
GeomAndNot　输出第一层中未被第二层覆盖的部分。你也可以理解为第一层减去第二层。有一点需要注意的是在这个命令中输入层的顺序不能搞错。



GeomXor 这个命令输出两层或多层之间非公有的部分。一般需要一或两个输入层。



GeomNot 输出输入层的反。只有一个输入层。



GeomCat 使所有的输入层连续。其输出包含所有的输入层。但不像 geomOr 使所有的层合并。可以有任意多个输入层。

关系命令

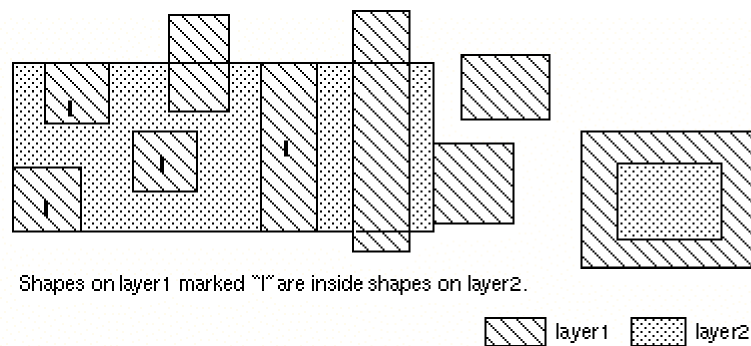
毫无疑问，关系命令必须有两个输入层次。所有的关系命令都将选择表达式中的第一输入层的整个图形（当然是满足条件的）。

大多数关系命令允许使用修改符。包括：连接符，限制符以及排除符。

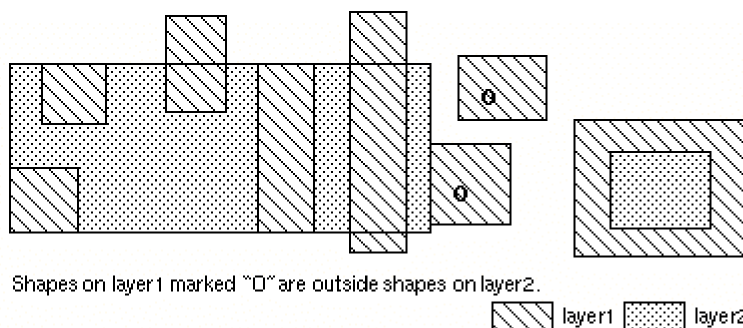
连接符的关键词为：diffNet 和 sameNet。前一个输出不同结点的图形。后一个输出相同结点的图形。

限制符的关键词为：keep 和 ignore。它们都需要指定被选定的图形个数的范围。

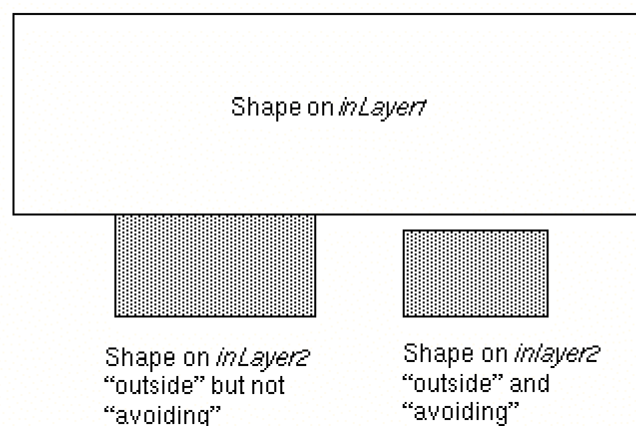
GeomInside　　选择完全处在第二输入层中的第一输入层。两层可以内切。不能使用限制符和排除符。



GeomOutside　　选择完全处在第二输入层之外的第一输入层。两层可以外切。不能使用任何修改符

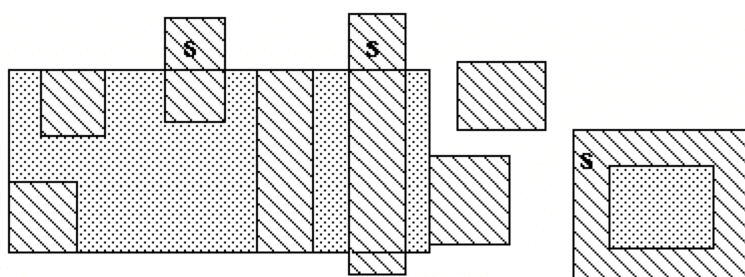


GeomAvoiding　　选择完全处在第二输入层之外的输入层。两层间不能外切。不能使用任何修改符

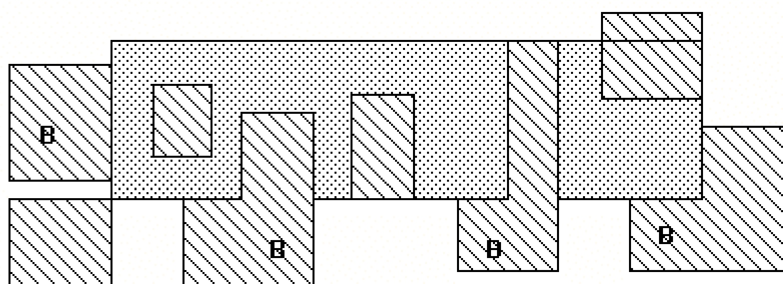


以下命令均可使用修改符

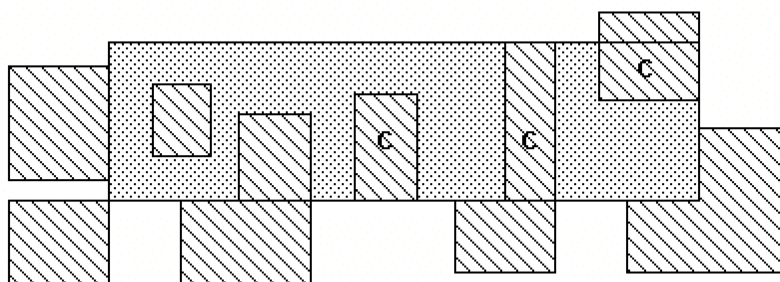
GeomStraddle　　选择的输入层只是部分被第二输入层所覆盖。



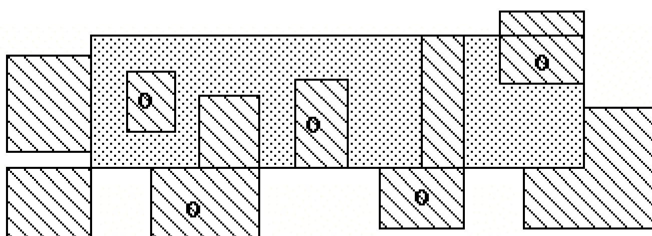
GeomButting 选择与第二输入层相外切的层次。



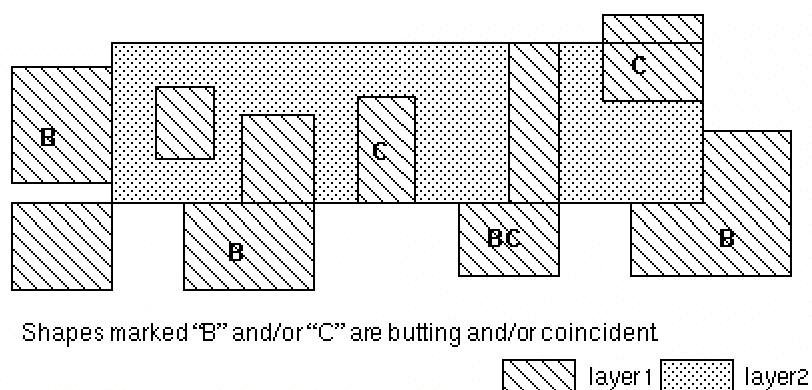
GomeCoincident 选择与第二输入层相内切的层次。



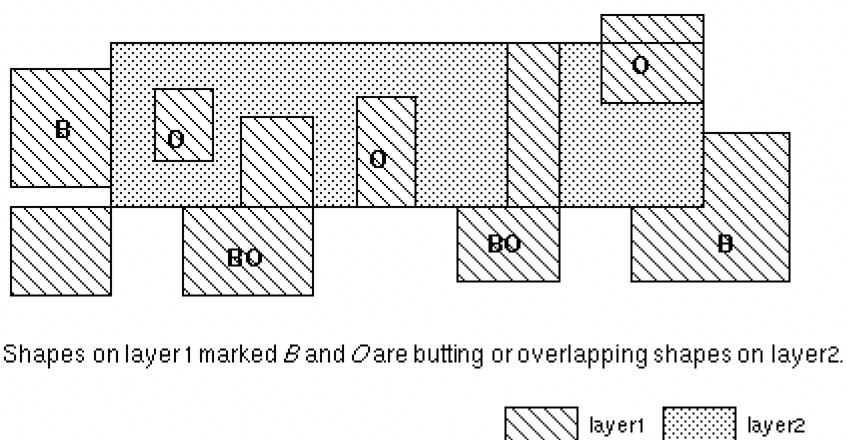
GeomOverlap 选择与第二输入层有公共面积的层次。



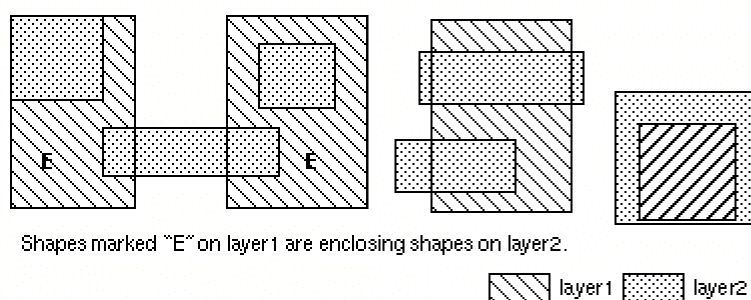
GeomButtOrCoin 选择与第二输入层外切或内切的的层次。



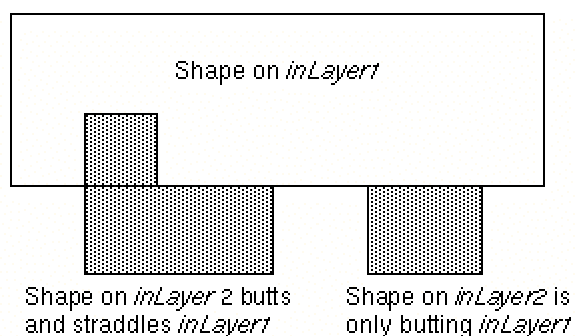
GeomButtOrOver　选择与第二输入层外切或相覆盖的层次。



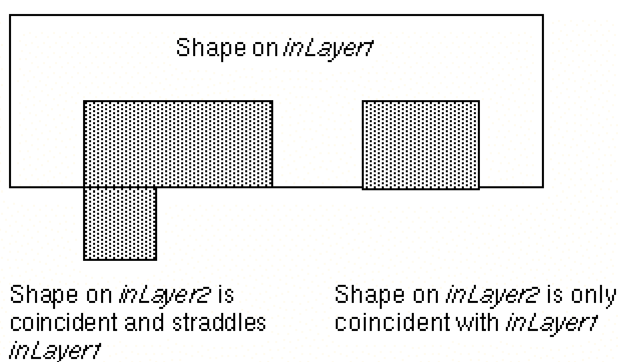
GeomEnclose　　选择完全包含第二输入层的层次，可以内切。



GeomButtOnly　　仅仅选择相互外切的层次.

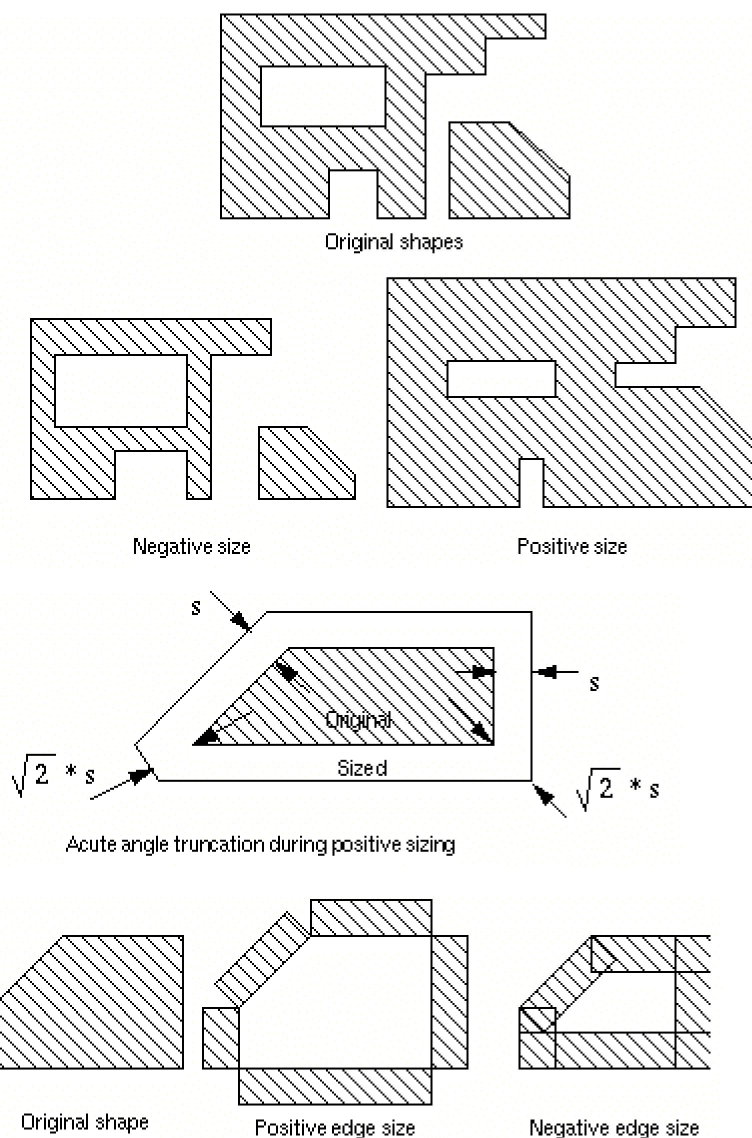


GeomCoinOnly　　仅仅选择内切的层次，第二层必须完全在的第一层内部。

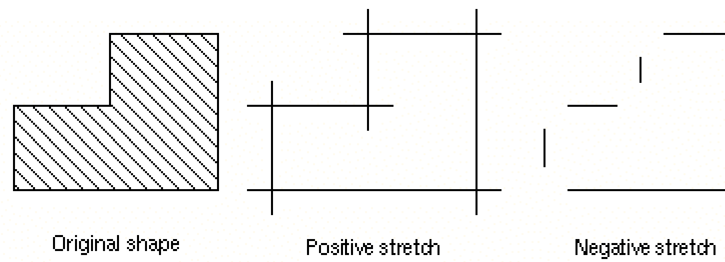


尺寸命令

geomSize　　按输入的数值扩张或收缩输入层。其中正值表示扩张，负值表示收缩。Edges 功能允许将图形的边按垂直方向舒张,正值表示向图形外部伸展,负值表示向图形内部伸展。有一点需要注意的是:所有的锐角在扩张时都会被截角。锐角在扩张时遵循与直角相同的规则。及伸展边扩张值的根号二。



GeomStretch　扩张或收缩输入层的边界。正值表示扩张，负值表示收缩。输出将会是边。



选择命令

根据输入层的特性，比如：形状，角度等等来选择输入层或其边界。

GeomGetRectangle　选择边平行或垂直于坐标轴的四边形。

GeomGetpolygon　与上一个命令搭配使用，选择上个命令没有选中的所有图形。

GeomGetVertex　根据顶点个数选择图形，顶点数目可由关键字 keep 和 ignore 来确定。

GeomGetAngledEdge　选择角度满足要求的边（注意其输出是边），但如果有关键词 fig，则输出包含选定边的图形。

GeomGetNon45　选择与坐标轴不平行，不垂直也不成 45 度角的边。

GeomGetEdge　根据边或边的一部分与其它边或图形的关系来选择。首先，要指出包含所需输出边的输入层，当然这个输入层可以是边或图形。然后，你可以使用以下操作符来选择所需的边：

Butting　选择输入层间相外切的边

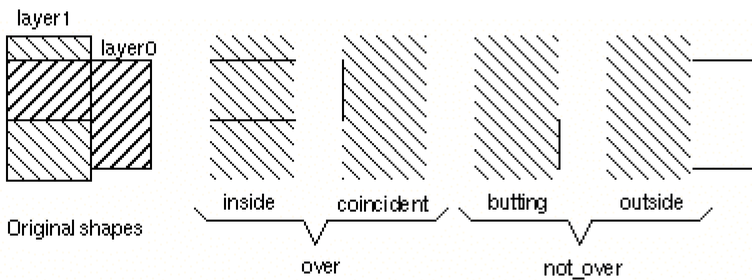
Coincident　选择输入层间向内切的边

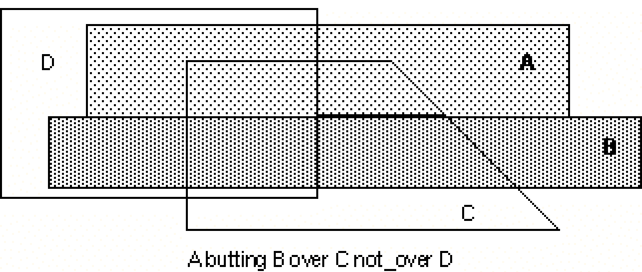
Outside　选择第一输入层中处在第二输入层外部的边

Inside　选择第一输入层中处在第二输入层内部的边

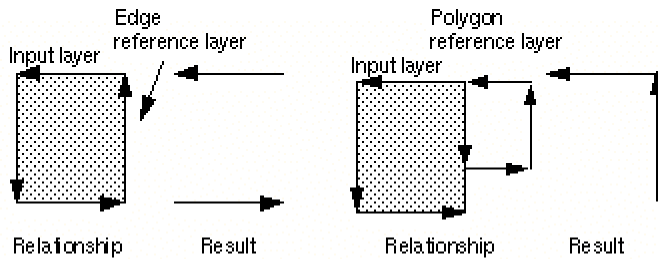
Not－over　为 outside 和 butting 的组合

Over　为 inside 和 coincident 的组合



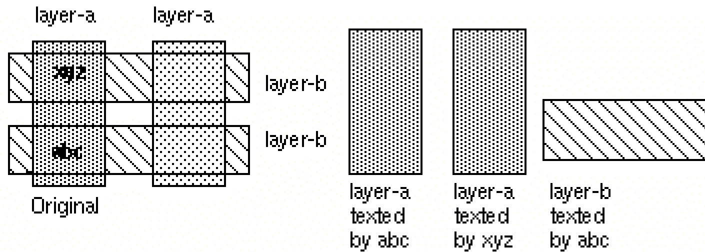


GeomGetAdjacentEdge　选择与其它参考边相临近的边

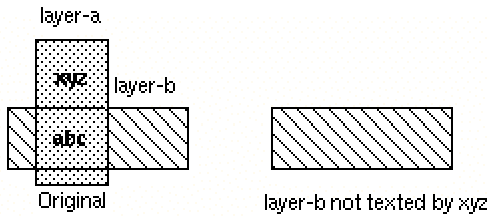


This function maintains the net numbers of shapes in the output layer.

GeomGetTexted　　根据文字标识来选择图形。所选图形必须包含文字标识。

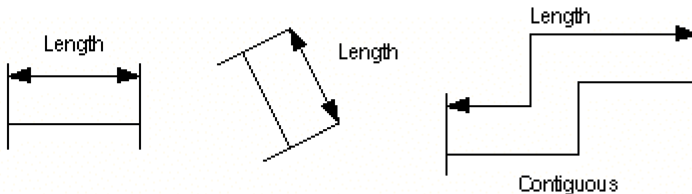


GeomGetUnTexted　　与上个命令正好相反。



GeomGetNet　　根据指定的节点名来选择图形。所有与这个指定节点相联系的层次都可以作为输入层。

GeomGetLength　　选择输入层中的边。选择根据是边的长度，这里所说的边可以是一条独立的边，也可以是折线边。



This function maintains the net numbers of shapes in the output layer.

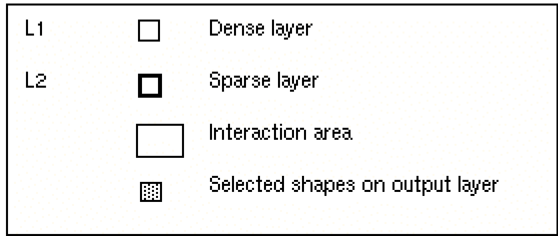
GeomHoled　　选择包含孔的图形（就像油炸圈饼一样）

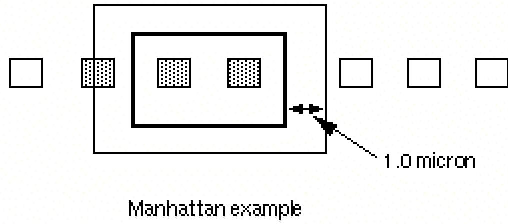
GeomGetNon90　　选择输入层中不平行于坐标轴的边。

焊点金属检查：

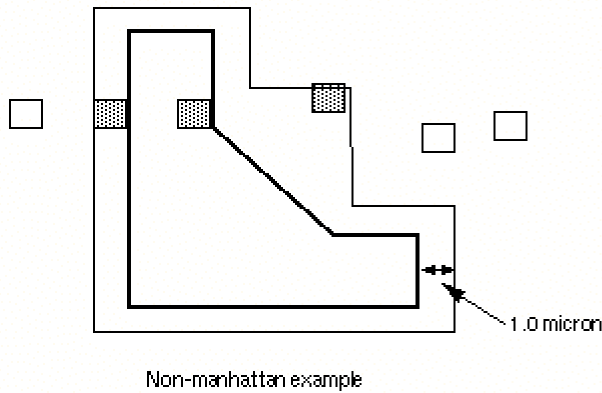
毫无疑问，对器件紧凑层的检查会比疏松层花费多得多的时间。举个例子，在焊点金属检查过程中，系统会检查所有金属的附近有无焊点，但是事实上是只在芯片边界上的金属边缘有焊点。这将会非常耗费时间，这里我们将使用 geomGetByLayer 命令来解决这个问题。以下步骤说明这个命令怎样运作：

1　一个边界将会包围在焊点层次周围，当然距离由我们来定。
2　接着，所有被这个边界包围或穿插的梯形金属层将会被选定。
3　最后，检查被选定的金属层与焊点间的间距（使用 DRC 命令）。





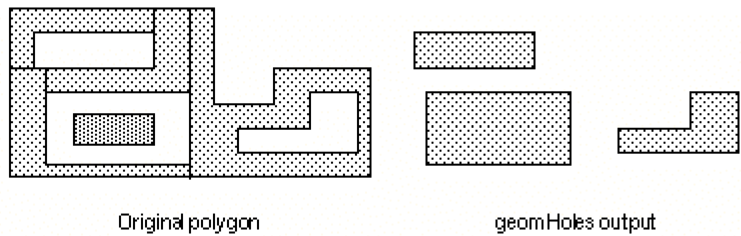
Manhattan example



Non-manhattan example

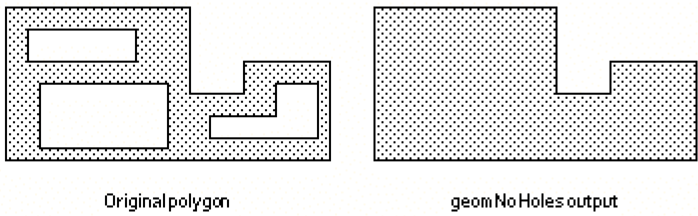
层生成命令

geomHoles　　　生成的图形具有“油炸圈饼”的几何形状。



Original polygon　　　geomHoles output

GeomNoHoles　　与上个命令相反



Original polygon　　　geomNoHoles output

GeomBkgnd　　　生成包含版图中所有图形的一个矩形。

存储命令

Savederived　　将衍生层存入库中相关的视图中去。当然你也可以指定输出层次，衍生层将会被存储在那里。

根据指定的关键字，衍生层可以被存储到某一个视图中去。这样的关键字有：lay_view,ext_view,cell_view,和 abs_view。从名字不难看出，它们分别代表版图，提取层，excell 和抽象视图。如果没有关键字，那么系统将默认是当前视图。有一点要注意的是，上述关键字的使用都必须在一个特定的上下文中，如：lay_view 应在 DRC 或 extraction 中。Ext_view 和 cell-view 必须在 extraction 中。而 abs_view 则只能在 abstract generation 中起作用。

这个命令的输出有两种格式：tile（梯形）和多边形。系统默认的是多边形格式。

这条命令也可以附加消息。但是消息两头必须有引号。可以在运行 DRC 的 explain 命令是看到这些消息。

CopyGraphics　　从当前分析层拷贝源层次到特定层中去。

在 extraction 时，源层次将被拷贝到 extracted view 中去。在 abstract generation 时，源层次拷贝到 abstract view 中。在分层提取中，如果使用到关键词 cell_view，系统会将指定层次从 " 分层提取中的最高层次(top_lever of hierarchy) " 中拷贝到 excell view 。但如果使用关键词 all，系统将会从设计版图的所有层次中搜寻指定层次并进行拷贝。

GeomErase　　从所有的视图中删去指定原始图形层上的全部图形。当然你也可以指定删除的范围。要特别小心的使用这个命令，因为他可能永久改变数据库。

DRC 命令

根据 drc 检查的类型，drc 命令可能使用一到两个输入层。输入层可以是原始图形层或是衍生层。

Drc 的检查结果可以被输出到输出层。而这些输出层又可以作为后续检查的输入。如果没有定义输出层，则检查结果会被输出到 marker 层去。

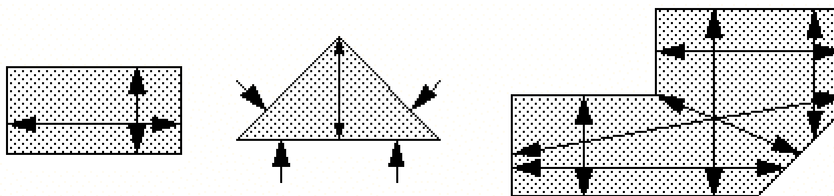
Fuctions－决定输入层被检查的方式。除了 area 外，所有的 function 都是基于边到边检查的。

每个 function 的使用都有一个范围。即有一个上限和下限。如果没有定义下限，系统会自动给出 " 大于等于 0 " 的保留值。

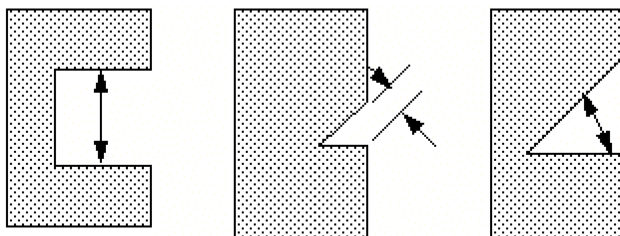
Modifiers－修改 functions 的属性。如果不冲突的话，可以同时使用多个 modifiers。

Functions：

Width　　检查同一输入层中内边线到内边线的距离



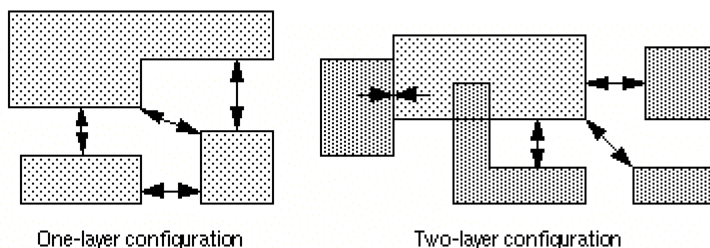
Notch　　检查同一输入层中外边线到外边线的距离



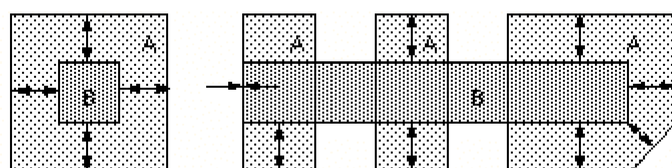
Adjacent edge checking is described in the description of the *drc* command.

Area　　检查单一输入层的面积

Sep　　检查第一输入层的外边线到第二输入层外边线的距离



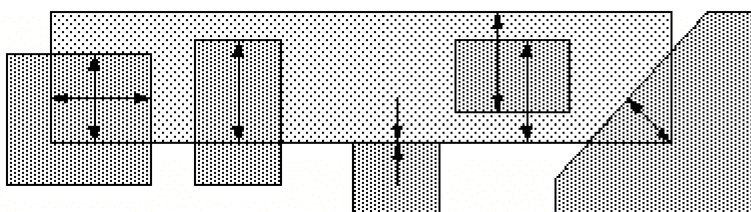
Enc　　检查第一输入层的内边线到第二输入层外边线的距离



drc(A B enc < 3)

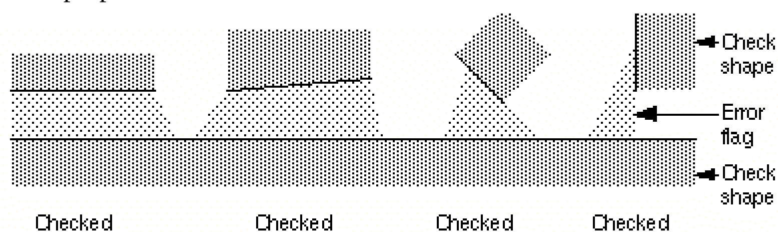
Enclosure of B inside A

Ovlp　　检查第一输入层的内边线到第二输入层内边线的距离

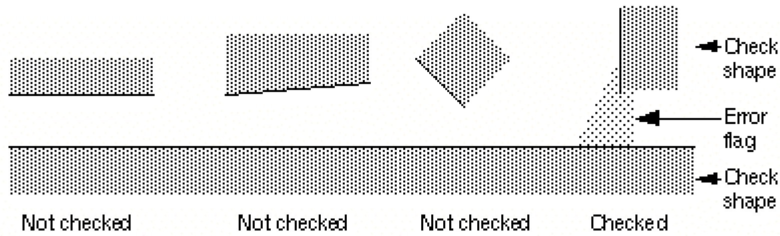


Modifiers:

With_perp　　除了标准的边界检查外还可以检查垂直边。

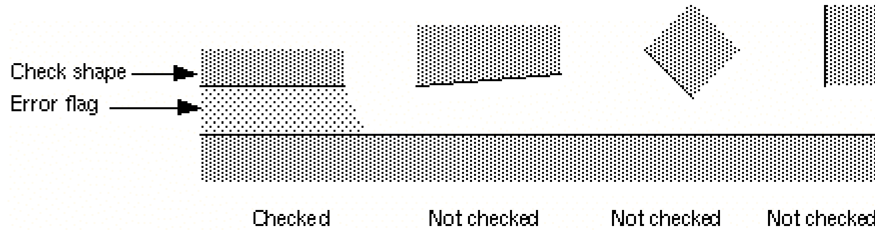


Only_perp　　只能进行垂直边的检查

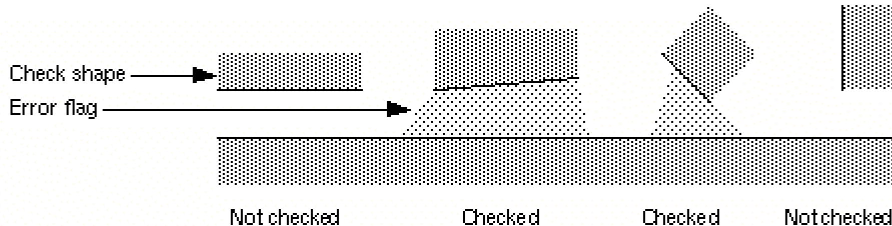


Use of this modifier excludes some other modifiers, such as *parallel*.

Parallel　　只能进行平行边的检查，所有 function 默认的边界检查为平行边或非平行边



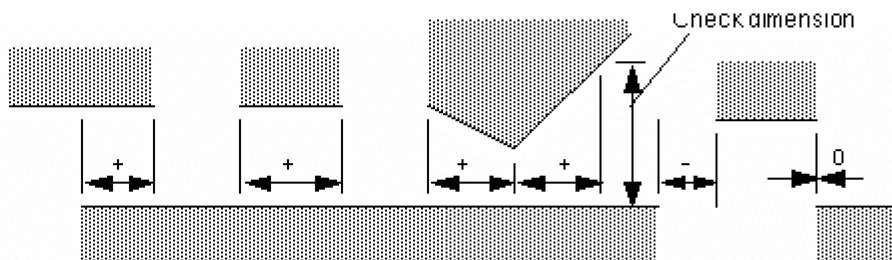
NotParallel　只能进行非平行边的检查



SameNet　　只检查连接在同一节点的层次，在这个命令里使用的层次必须在前面的 geomConnect 命令中出现过

DiffNet　　只检查连接在不同节点的层次，在这个命令里使用的层次也必须在前面的 geomConnect 命令中出现过

App　　只检查值确定的投影边。无论是正值、负值或是零都必须说明



Key: + positive app. - negative app. 0 zero app.

Opposite　　只检查两图形的相对边。将输出修整为正的并置。

Length，lengtha，Lengthb　　只计算长度符合指定长度限制的边的长度。Length 在检查两层关系时，两条边都计算。Lengtha 和 lengthb 则分别代表第一层和第二层的边

Fig，figa，figb　　fig 在两层检查中两层的图形都输出。而在一层检查中输出第一层的图形；figa 和 figb 在两层检查中分别输出的第一层和第二层的图形

Edge，edgea，edgeb　　edge 在两层检查中两层的边界都输出。而在一层检查中输出第一层的边界；edgea 和 edgeb 在两层检查中分别输出的第一层和第二层的边界

Raw　　在层次合并之前检查每一层。

Message　　是一串处在引号中的字符串，用于错误检查中提示

方形边角检查

SquareGrow　　在图形周围添加一个直交的检查图形，即将原图形沿 x 和 y 方向伸展指定的长度。形象的说就是在原图形外加一个封套

NormalGrow　　在原图形周围添加一个扩张的图形。在使用这个命令时，有一个特殊情况：如果图形的挖槽（notch）长度小于两倍的扩张长度，将会在槽处出现冲突。这一点我们要注意到。

大多数 DRC 规则可以分为以下三部分：

1 电气规则要求的指定材料间的露头（enclosure）或间距（sepration）
2 在生产时可能存在的掩膜错位
3 生产过程中材料的过搀杂或是迁移

上述的方形边角检查可以使对非投影的折角描述更加精确。

Soft-connect 检查

GeomStamp　　检查阱是否被错误的使用来作为传导通道。geomstamp 必须要放在 geomConnect 命令之后而且 stamping 层(第二层)必须在 geomConnect 中出现过。

在复杂的单元覆盖情况下使用分层 DRC 检查来提高效率：

hier？　允许在分层 DRC 检查时使用不同的规则，它包括以下分支句子：

currentCell？只作当前单元的检查

topcell？　　只作最高层单元（top－level cell）的检查

当你在 DRC 检查中选择了分层检查模式，那么所有在 hier？分支语句中的命令将会被执行。

在 DRC 中运用层次属性：

层次属性：

spacingRules 定义一层或是两层的无序属性。DRC 保留的属性有：minWidth，minSpacing，和 minNotch。

OrderedSpacingRules 定义指定顺序的两层的属性。DRC 保留的属性有：minEnclosure，minSpacing 和 minOverlap。

变量赋值：

DRC 在 technology file 中可以设定变量。一般是在 technology file 中的 controls class 中。

DRC 中的参考值：

CheckLayer　　进行一层或是两层的检查.

CheckAllLayers 在所有层上进行 DRC 检查。当然你可以指定某一层不被检查.

TechGetSpacingRule 从单层属性中获取值

TechGetTechFile　　获取设计的 technology file 信息